DYNAMIC DATAWINDOWS

PowerBuilderJournal.com

U.S. $15.00 (CANADA $16.00)

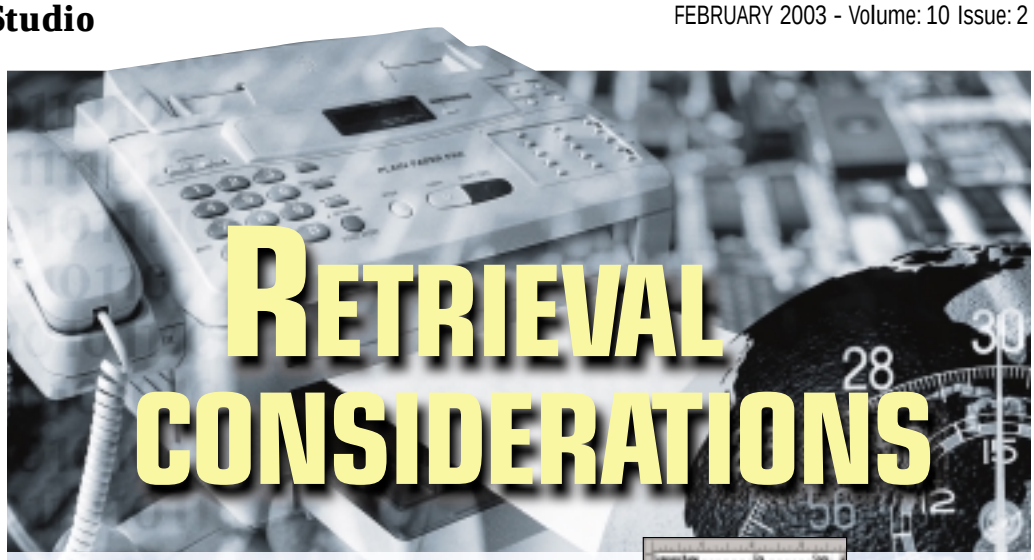
DEVELOPER'S
PowerBuilder Journal

Enterprise Application Studio

FEBRUARY 2003 - Volume: 10 Issue: 2

www.PowerBuilderJournal.com

Announcing...
web services EDGE conference&expo
March 18-20, 2003
Boston, MA
HYNES CONVENTION CENTER
SEE PAGE 9 FOR DETAILS



RETRIEVAL CONSIDERATIONS

From the Co-editors
Information Liquidity
by John Olson pg. 3

Soft Skills
by Bob Hendry pg. 4

PB Utility
Wallpaper Setter in PowerBuilder
by Atul Kane pg. 20

Industry Announcements
by Bruce Armstrong pg. 34

Focus: DataWindow Reporting
Composite versus nested reports — Bob Hendry

PBDJ Feature: Dynamic DataWind… Restoration and Retri…
to existing applicati… — Berndt Hamboeck 16

…ions: Mobile Development
Gain a competitive edge in business — Scott Heffron 22

PBDJ Feature: PowerBuilder, Meet the JCE!
Call cryptographic methods PART 2 — Jason Weiss 24

PowerBuilder 101: Dynamic Report Module Basics
Take your PB apps to the next level — David Hart 28

Book Excerpt: Using Oracle Stored Procedures in Your DataWindow — Bruce Armstrong 30

SPECIAL OFFER!
WEB SERVICES EDGE 2003 EAST
INTERNATIONAL WEB SERVICES CONFERENCE & EXPO
SAVE $400
IF YOU REGISTER BEFORE FEBRUARY 28, 2003

…AILS ON PAGES 32 & 33

RETAILERS PLEASE DISPLAY
UNTIL APRIL 30, 2003
$15.00US $16.00CAN
02>
0 09281 03424 7

SYS-CON MEDIA

Amyuni Technologies, Inc.

www.amyuni.com

PowerBuilder Developer's Journal

EDITORIAL ADVISORY BOARD
BRUCE ARMSTRONG, MICHAEL BARLOTTA, ANDY BLUM, RICHARD BROOKS, KOUROS GORGANI, BAHADIR KARUV, PH.D., BERNIE METZGER, JOHN OLSON, SEAN RHODY

COEDITOR-IN-CHIEF: JOHN OLSON
COEDITOR-IN-CHIEF: BOB HENDRY
EXECUTIVE EDITOR: NANCY VALENTINE
ASSOCIATE EDITOR: JAMIE MATUSOW
ASSOCIATE EDITOR: GAIL SCHULTZ
ASSOCIATE EDITOR: JEAN CASSIDY
ASSOCIATE EDITOR: JENNIFER STILLEY
ONLINE EDITOR: LIN GOETZ
TECHNICAL EDITOR: BERNIE METZGER
NEWS EDITOR: BRUCE ARMSTRONG
DATAWINDOWS EDITOR: RICHARD BROOKS

WRITERS IN THIS ISSUE
BRUCE ARMSTRONG, BERNDT HAMBOECK, DAVID HART, SCOTT HEFFRON, BOB HENDRY, ATUL KANE, JOHN OLSON, CHARLES SILBERGLEITH, IGOR SKOKIN, JASON WEISS

SUBSCRIPTIONS
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS, PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT

SUBSCRIPTION HOTLINE: 888 303-5282
COVER PRICE: $15/ISSUE
DOMESTIC: $149/YR. (12 ISSUES) CANADA/MEXICO: $169/YR.
OVERSEAS: BASIC SUBSCRIPTION PRICE PLUS AIRMAIL POSTAGE (U.S. BANKS OR MONEY ORDERS). BACK ISSUES: $12 U.S., $15 ALL OTHERS

PRESIDENT AND CEO: FUAT A. KIRCAALI
COO/CFO: MARK HARABEDIAN
VICE PRESIDENT, BUSINESS DEVELOPMENT: GRISHA DAVIDA
SENIOR VP, SALES & MARKETING: CARMEN GONZALEZ
PRODUCTION CONSULTANT: JIM MORGAN
VICE PRESIDENT, SALES & MARKETING: MILES SILVERMAN
ACCOUNTS RECEIVABLE KERRI VON ACHEN
FINANCIAL ANALYST: JOAN LAROSE
ACCOUNTS PAYABLE: BETTY WHITE
ADVERTISING DIRECTOR: ROBYN FORMA
ADVERTISING ACCOUNT MANAGER: MEGAN RING-MUSSA
ASSOCIATE SALES MANAGER: CARRIE GEBERT
ASSOCIATE SALES MANAGER: ALISA CATALANO
ASSOCIATE SALES MANAGER: KRISTIN KUHNLE
PRESIDENT, EVENTS: GRISHA DAVIDA
CONFERENCE MANAGER: MICHAEL LYNCH
REGIONAL SALES MANAGER: MICHAEL PESICK
REGIONAL SALES MANAGER: RICHARD ANDERSON
ART DIRECTOR: ALEX BOTERO
ASSOCIATE ART DIRECTOR: LOUIS F. CUFFARI
ASSOCIATE ART DIRECTOR: RICHARD SILVERBERG
ASSISTANT ART DIRECTOR: TAMI BEATTY
VICE PRESIDENT, INFORMATION SYSTEMS: ROBERT DIAMOND
WEB DESIGNER: STEPHEN KILMURRAY
WEB DESIGNER: CHRISTOPHER CROCE
CUSTOMER SERVICE REP: MARGIE DOWNS
JDJ STORE: RACHEL MCGOURAN

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9600
SUBSCRIBE@SYS-CON.COM

POWERBUILDER DEVELOPER'S JOURNAL (ISSN#1078-1889) is published monthly (12 times a year) for $149 by SYS-CON Publications, Inc.,135 Chestnut Ridge Rd., Montvale, NJ 07645
Periodicals Postage rates are paid at Montvale, NJ 07645 and additional mailing offices.
POSTMASTER: Send address changes to: POWERBUILDER DEVELOPER'S JOURNAL, SYS-CON Publications, Inc., 135 Chestnut Ridge Rd., Montvale, NJ 07645

© COPYRIGHT
Copyright © 2003 by SYS-CON Publications, Inc. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system, without written permission. For promotional reprints, contact reprint coordinator Carrie Gebert. SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.

All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies. SYS-CON Publications, Inc., is not affiliated with the companies or products covered in PowerBuilder Developer's Journal.

JOHN OLSON, EDITOR-IN-CHIEF

Information Liquidity

DataWindow liquidity

The dizzying whirlwind of terms we are bombarded with is…well…dizzying. Apparently, the age-old term "data flow" was no longer capable of describing what we want to do with our data. Or maybe Sybase wants to take our data to the next level and make it really valuable, because we haven't been squeezing the last bits of efficiency and value out of it?

Recently, Sybase began using a new term, *information liquidity*. I hate new terms created solely to grab our attention, but I hope Sybase trademarked this one because I love it. At first I thought liquidity was an adjective to help describe really fast and flexible data flow. But this term also brings new meaning to the table. Webster's describes "liquidity" as "the quality of being readily convertible to cash." Sybase describes information liquidity as "the efficiency at which businesses transform enterprise data into economic value" and says "In the same way that ATMs redefined, expanded, and improved cash liquidity for consumers, Information Liquidity redefines the Speed, Flow and Quality of data within the enterprise." Sounds great, though maximizing data availability and flow has been a goal of companies for decades. Is it possible Sybase isn't just coining a new term to describe existing ideas, but is going to provide a mechanism that will revolutionize data quality, management, and use?

Sybase's goal is to provide the best tools for viewing and aggregating enterprise data, from a broad variety of sources, with maximum flexibility and power. Its enterprise portal and data warehousing tools are the main components of a diverse set of products that are aimed at helping companies achieve this goal. Undoubtedly, just like openness has been a key aspect of each of Sybase's products, so too is intercommunication and interoperability, which is necessary to achieve the level of liquidity Sybase is endeavoring to provide. Imagine that a company's systems, old and new, seamlessly share data, as well as make it available for viewing worldwide via an enterprise portal. BEA's recent announcement of "Liquid Data" indicates that a race is on to provide the best tools to help companies achieve that goal.

Recently Sybase launched a free online Information Liquidity assessment tool to help businesses "transform data into economic value." To assess your company, or find out more about this concept, go to www.informationliquidity.com.

DataWindows

Have you lost your excitement for the DataWindow? Has it become a boring, blasé topic? Granted, we've been publishing articles and books about the DataWindow since 1994 so it isn't exactly a new topic. However, it is the premier feature of PowerBuilder and it keeps getting more and more powerful. Every year when we put together the editorial calendar for ***PBDJ,*** we plan on having an issue that focuses on the DataWindow. It's a feature that interests all our readers, whether developing client/server, *n*-tier, or Web.

This year, with the release of PowerBuilder 9 nearly upon us, the excitement about the DataWindow is mostly about the new XML features. Certainly, it will catch the interest of Web developers, but it is also a valuable feature for *n*-tier and client/server development. That point is often overlooked because people don't understand the value of XML in general nor of the DW XML features. At a minimum, it gives us an additional format for storing our DataWindow data and presentations. Even more, it's simply a string, but as XML it will be self-descriptive, and in some situations will be usable by non-PB clients. It'll be like the PSR but mostly open, rather than proprietary, and won't require a PB plug-in, DLL, or exe in order to read it. Since a string is a basic data type, imagine being able to pass the DW, including presentation and data, between clients of widely differing types, written in any language, on any platform – COM, CORBA, C, Java, PB, JavaScript, application server component, Windows client, etc. That's DataWindow liquidity! ▼

john@sys-con.com

AUTHOR BIO
John Olson is principal of Developower, Inc., a consulting company specializing in software solutions using Sybase development tools. A CPD professional and charter member of TeamSybase, he is a coauthor of SYS-CON's Secrets of the PowerBuilder Masters books, and co-editor and author of two upcoming PB9 books.

Soft Skills

WRITTEN BY
BOB HENDRY

Tech workers, stunned by rounds of layoffs in the aftermath of the dot-com crash and the downturn in the economy, continue to hit a wall in a labor market that has fewer jobs, lower salaries, pickier bosses, and a new job requirement: soft skills.

Soft skills? Today, tech workers who do not expand their experiences and interests outside the world of computers and labs may get passed over for those who also have good communication and leadership skills and can work in teams. Employers are adamant that they have to have soft skills. They want to see people who are better rounded, not just absorbed in the techie thing.

Five years ago employers were looking for code warriors – employees who knew what all the acronyms meant. That has all changed. Today they're looking for employees who have the ability to work in teams, communicate effectively, and be aware of the social and cultural context of what they're doing and how it enables them to relate to others. A couple of years ago, employers would have hired somebody with great technical skills; today, potential employees have to have more than that. For tech workers accustomed to winning over employers with their beefy résumés, the demand for soft skills could be a challenge. The days of head-down coding are over – at least for now. Code jockeys are so last century.

During the bubble and the ensuing war for talent, some hiring managers wanted to get a person in front of the computer as soon as possible. Now the motto is to get the right person for the right job at the right time. Things are bound to slowly pick up.

Still, we may be in for a long winter. The tech industry is on life support nationwide. Large employers that laid off thousands aren't expected to hire again anytime soon. And much of the growth in technology is in areas such as security, defense, and biotechnology, which may require a set of skills many of us do not yet have.

There are glimmers of hope. The government, pharmaceutical companies, and some banks have been able to pick up highly skilled tech workers they couldn't afford before, and the information sector, including broadcasting companies, call centers, and managed data firms, are still hiring.

Some of us have turned to other careers or taken cover in graduate school; many of us are waiting out the storm by taking lower-paying or temporary jobs, or going into business for ourselves.

We were among the first professionals to feel the impact of the economic slowdown starting with the dot-com bust late in 2000. Before that, we were the darlings of the working world, commanding signing bonuses, high salaries, and on-the-job perks. The technology workforce is suffering more than anybody as the labor market we're experiencing has affected the tech industry the most. The unemployment rate among tech professionals is 12–20% depending on your market.

In my market (Chicago), recent tech professionals have been especially hurt by massive layoffs at large companies such as Motorola, Sears, Tellabs, Arthur Andersen, and Lucent. Because there is so much overcapacity (supply) of labor, salaries and contract rates have fallen. With so many experienced professionals on the market, there is almost no work for recent graduates or for those with little or no experience.

If you are a recent graduate and are lucky enough to find any tech job, starting pay is stagnant – starting salaries for computer science and engineering graduates range from $39,000 to $45,000. Companies that are hiring not only prefer experienced candidates but also people who are well rounded. These days, it takes more than a college degree, up-to-date skills, and a pair of Doc Martens to land a job. Employers still place a high value on expertise, prompting many tech professionals to earn certification in the most popular tech skills, with Microsoft certification being one of the most popular. Certification helps employers differentiate one candidate from another.

However, even experience and certification will not guarantee a job in today's climate. Like other industries, the health of the tech world hinges on employer confidence. Some experts estimate that the tech job market will lag for another 18 to 24 months before employers start hiring again.

Temporary or contract work is still out there at a reduced rate. When things improve, these workers will be among the first to be called back. Since employers are nervous about making long-term commitments to full-time employees, they are increasingly farming out small projects to free agents.

Don't give up. Keep your résumé up to date. Focus on the "whys" and not the "hows". As one headhunter told me, you should be able to explain why your past project was able to use XML to streamline data access and provide value to its users, and you should be able to explain it in nontechnical terms. Chances are that the person interviewing you could care less if you know a SAX parser from a DOM parser. He or she wants to know how you can take your knowledge and experience and communicate it to a team – both technical and nontechnical. ▼

bob@sys-con.com

AUTHOR BIO

Bob Hendry is a PowerBuilder instructor for Envision Software Systems and a frequent speaker at national and international PowerBuilder conferences. He specializes in PFC development and has written two books on the subject, including Programming with the PFC 6.0.

Sybase, Inc

www.sybase.com

DataWindow Repo

Composite

WRITTEN BY BOB HENDRY

In my early PowerBuilder days, I was very impressed with how easy it was to create and format DataWindows for either online or printed reports. At its easiest, a simple print method is required to send a "What you see is what you get" version of the DataWindow to the printer. Even when the printed version of a DataWindow is different from the visual one, a simple ShareData() function would be all that is required.

What took a while, however, was learning how to display and print reports that either had a master/detail relationship or no relationship at all. In short, what if I need to create or print a report that's based on different SQL statements? There are two solutions to this problem:

1. Create a master (or base) DataWindow, then add additional nested DataWindows (nested reports).
2. Create a report by using the DataWindow composite presentation style.

Nested Reports

The first way to include a report within a report is to add one to an existing DataWindow. Nesting reports is a common practice when dealing with a master/detail relationship. Adding a nested report is straightforward: within the master DataWindow select the report icon from the control palate, then drop it in the detail band.

Figure 1 contains a master DataWindow that displays the columns from the department table. Within the detail line on the left-hand side is a nested report that contains a list of all employees who work within the department displayed on each detail line. The nested DataWindow takes the department ID for a retrieval argument. A design time view is displayed in Figure 1.

When it was created, the nested DataWindow d_chap_emps was defined to accept a retrieval argument – the department ID – and then display employee information for that department. To establish the link between the master DataWindow and the nested one, supply the arguments property of the nested DataWindow with an appropriate value, in this case the value of the dept_id column. Now when the master/detail window is retrieved, the department ID will be passed as a retrieval argument to the nested DataWindow, in turn displaying all employees for that department.

Nested reports don't always have to exist in the detail band. Figure 2 displays a master DataWindow grouped by the state column. The group band contains a nested report that has been set up to accept state as an argument. If you place a nested report in any band except the detail band, the autoheight property of the nested report becomes meaningless as the detail band is the only DataWindow band that will adjust its size to fit its contents. Also, note that when adjusting the height or width of a nested report or band, it may be necessary to save changes, exit the DataWindow painter, and then reenter to see the formatted changes.

The previous examples demonstrate how to use a nested report when a master/detail relationship exists. Actually, no relationship needs to exist at all. A nested report can exist in any band within the DataWindow. Where a nested report exists and what it displays is up to you. The DataWindow in Figure 3 contains a nested report in the header band. As long as the nested report does not require a retrieval argument, this is perfectly valid.

When placing nested reports in bands other than the detail band there are times when you may want to repress the display of the nested report. For example, rather than displaying the nested report at the top of every page, only show it on the first page. This will be discussed later in this column series when I cover DataWindow bands.

NESTED REPORTS – HOW DOES THE DATA GET RETRIEVED?

When running or previewing a DataWindow containing a nested report, the actual number of database retrievals is determined by which band the nested report is in. First, the data is retrieved for the base DataWindow. If the detail band contains a nested report, each row of the detail band will retrieve the nested report that is associated with it. If the detail band contains

100 lines, 100 retrieves will be performed for the nested report. Nested reports not in the detail band are retrieved once for however many times that band exists.

NESTED REPORTS – ACCESSING OBJECTS

Since a nested report is not a DataWindow child, objects within the nested report cannot be accessed with a getchild() method. To set object properties (including expressions), dot notation must be used. The syntax is:

```
dwcontrol.Object.nestedreportlname { [row ]
} .Object.dwcontrolname.
      property { .property } { = value }
```

For example, the following code will set the expression for the salary column within the nested report. The syntax below will set the expression for all instances of the nested report dw_emps:

```
dw_1.object.dw_emps.Object.salary.color =
'255 ~t If(salary > 50000, 255, 0)'
```

It's possible to access and change properties in a nested report object if they are dependent on a value in the base DataWindow. The following code example loops through all rows in the base DataWindow; if the department ID is equal to 100, the nested report for that specific row is accessed. An expression is then applied to the salary column. Notice the syntax below the row of the nested report is being referenced.

```
integer li_row
FOR li_row = 1 to dw_1.RowCount()
 IF
String(dw_1.Object.dept_id.Primary[li_row])
= "100" THEN

dw_1.Object.dw_nested_rpt[li_row].Object.sal
ary.Color = &
  '255 ~t If(salary > 60000, 255, 0)'
 END IF
NEXT
```

If the report does not have a row associated with it, no row specification is needed when accessing the nested report. The following code is valid if accessing a nested report in the header, footer, or summary band within the base DataWindow. If the report is in a header of a trailer band, the row number for the band must be specified.

```
dw_1.object.dw_nested_report.Object.state.co
lor = &
  '255 ~t If(state <> ~'IL~', 255, 0)'
```

NESTED REPORTS AND THE GETCHILD() METHOD

Earlier, I mentioned that report objects nested within a DataWindow are not DataWindow children, thus any call to getchild() will fail. There is a way around this. In the source editor, open the base DataWindow and change "processing=0" (or whatever it is) to "processing=5". This is the actual trick to change the base DataWindow's presentation style to "composite". The getchild function can now be used to access the nested report.

This is a cool trick because it opens up all methods that can be called for DataWindow children. A drawback of this method is that since the base DataWindow's presentation style has been changed to composite and composite windows have no data source, the base DataWindow's data source cannot be changed unless processing is changed back from "5". Also, it should be pointed out that the above technique is not supported by Sybase and should be used with extreme caution.

Composite Reports

An alternative to using nested reports is the use of the composite DataWindow presentation style. A composite DataWindow is really a collection of unrelated DataWindows (see Figure 4). Composite DataWindows by nature do not have a data source; the data comes from the data source of the composite DataWindow objects.

RETRIEVAL CONSIDERATIONS

Even though a composite DataWindow has no data source, if all enclosed DataWindows have no retrieval arguments, all may be retrieved via one Retrieve() function. The following code will retrieve all DataWindows within a composite report.

```
dw_1.SetTransObject(SQLCA)
dw_1.Retrieve()
```

This approach works well when all nested DataWindows within the composite report have no retrieval arguments. If retrieval arguments are needed, there are two ways they can be specified. First, they can be supplied to the DataWindow child. For example, let's say the composite DataWindow in Figure 4 takes two retrieval arguments:

Nested DataWindow:	Retrieval Argument:
d_cust_list	State (String)
d_product_graph	Region (String)

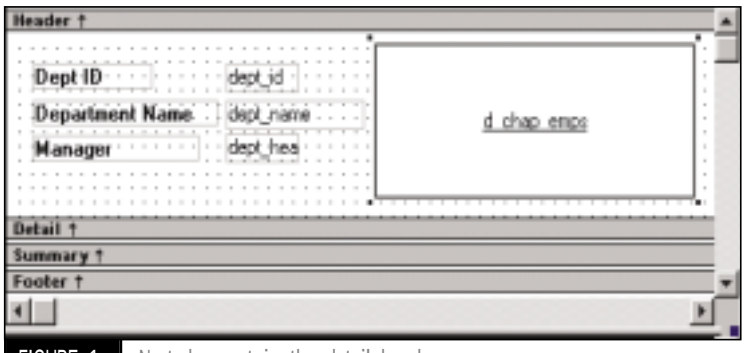


FIGURE 1 Nested report in the detail band

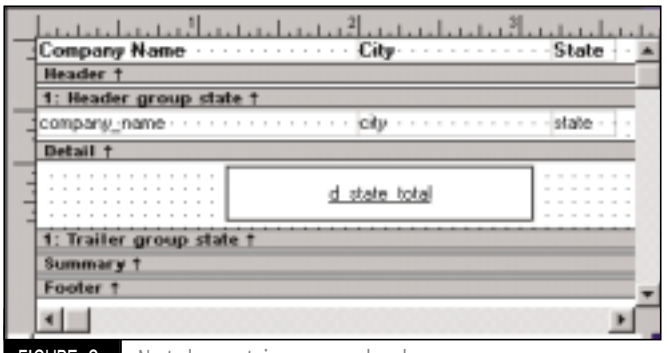


FIGURE 2 Nested report in a group band

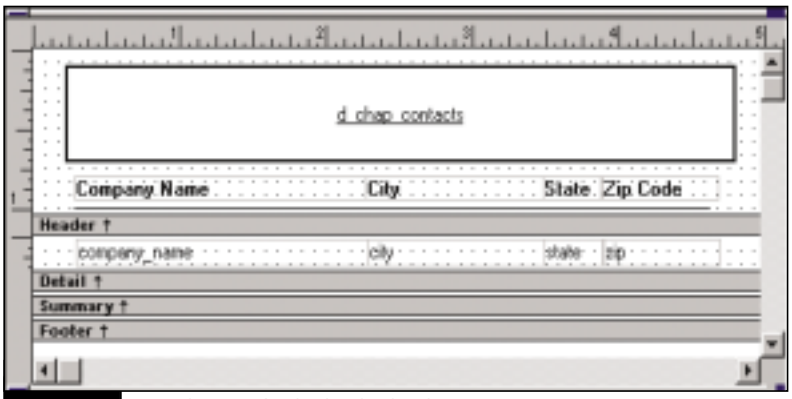


FIGURE 3 Nested report in the header band

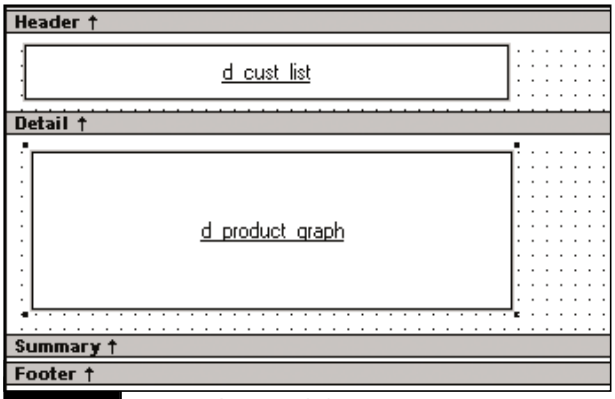


FIGURE 4 A composite DataWindow

The code to retrieve them would be:

```
DataWindowChild    ldwc_cust
DataWindowChild    ldwc_product
dw_1.GetChild("dw_cust",ldwc_cust)
dw_1.GetChild("dw_product",ldwc_product)
ldwc_cust.SetTransObject(SQLCA)
ldwc_product.SetTransObject(SQLCA)
ldwc_cust.Retrieve("IL")
ldwc_product.Retrieve("Central")
```

Notice when obtaining a handle to the DataWindow child, the actual name of the nested report is specified, not the DataWindow object associated with it. Furthermore, for the sake of brevity and readability, return codes were not checked in the this code example. Because the nested DataWindows are truly DataWindow children, a GetChild() method can be called to obtain their handle. Any method that is valid for a DataWindow child may now be used.

If you don't want or need to obtain the nested reports as DataWindow children, there's a way in which they may be retrieved directly – without obtaining their child handle. Retrieval arguments can be specified within the DataWindow Painter as a general property of the composite DataWindow (see Figure 5).

Now that the retrieval arguments are a general property, they need to be plugged into the nested DataWindows. From within the DataWindow Painter, right-click on the nested report and supply its retrieval argument under the General Properties tab (see Figure 6).

This figure ties in the "state" retrieval argument for the nested report to the "state" retrieval argument that was supplied in Figure 5 as a general property of the composite DataWindow. The same steps should be taken to supply the "region" argument to the other nested report, but the steps are redundant and so won't be repeated here.

The final step in this method of composite DataWindow retrieval is the actual code:

```
dw_1.SetTransObject(SQLCA)
dw_1.Retrieve("IL","Central")
```

Notice that when retrieving nested reports (for composite DataWindows) in this fashion, you don't have to programmatically obtain the DataWindow child.

DYNAMIC NESTED REPORTS

The DataWindow object of the nested report does not have to be defined at design time – it can be dynamic. The following PowerScript determines the DataWindow object as well as any arguments.

```
dw_1.SetTransObject(SQLCA)
If gs_userid = "MANAGER" Then
 dw_1.Modify("dw_product.DataObject =
~"d_product_graph~"")
Else
 dw_1.Modify("dw_product.DataObject =
~"d_product_by_region~"")
 dw_1.Modify("dw_product.Nest_Arguments =
((~"'Central'~"))")
End If
dw_1.Retrieve()
```

In this code, the nested report named dw_product will contain the DataWindow Object d_product_graph if the user logged on is MANAGER. All other users will see the DataWindow object d_product_by_region, which takes a string argument in which we are supplying the value "Central".

Notice how the tilde character was used to denote the double quote character within a string. The DataObject property for the nested report looks and acts the same way as it would for a DataWindow. What is noteworthy is the Nest_Arguments property for the nested report. This property is used to supply either a literal value, expression, or a column value as a retrieval argument(s) for a nested report. The number of values provided in the list has to match the number of retrieval arguments defined for the nested report. Even though this example is in the composite report section, it will work for nested reports as well – that is, when a DataWindow is not of the composite report type but does include nested reports. The syntax for supplying dynamic retrieval arguments via the Nest_Arguments property follows:

```
( ("arg1") {,("arg2") {,("arg3") {,... } }
} )
```

Notice that this list is not a quoted string. The whole list is surrounded by an open and a close parenthesis, respectively. Each argument is also surrounded by an open and a close parenthesis. Finally, the argument is enclosed by quotes. Also, the way the argument is treated by PowerBuilder differs depending on how it's presented. Sound confusing? It is. The following examples should help simplify this.

USING EXPRESSIONS OR COLUMN VALUES

When the argument is equal to a database column (in a nested report that is not a composite DataWindow), its value is supplied in double quotes. The following code example passes the current value of the cust_id column as a retrieval argument.

```
dw_1.Modify("dw_product.Nest_Arguments =
~"cust_id~"")
```

USING STRING VALUES

String values must be entered in single quotes within the double quotes:

```
dw_1.Modify("dw_product.Nest_Arguments =
((~"'Central'~"))")
```

USING NUMERIC VALUES

When using numbers, omit the single quotes. Notice that the literal value 100 is still surrounded by double quotes. Remember the double quotes denote the value for the retrieval argument for the nested report; they do not denote a "string".

```
dw_1.Modify("dw_product.Nest_Arguments =
((~"100~"))")
```

PROMPTING THE USER FOR RETRIEVAL ARGUMENTS

When more than one argument is needed, they are separated by commas. If not enough (or too many) arguments are defined, a runtime error will occur. If you want to prompt the user to enter the retrieval arguments at runtime, specify and empty parentheses like:

```
dw_1.Modify("dw_product.Nest_Arguments =
()")
```

HOW DYNAMIC CAN WE GET?

When using nested reports in a composite

International Web Services Conference & Expo

Web Services Edge 2003

CONNECTING THE ENTERPRISE WITH WEB SERVICES, JAVA, XML, AND .NET

March 18-20, 2003
Boston, MA

web services EDGE conference&expo

JDJ EDGE conference&expo

XML EDGE conference&expo

.NET EDGE conference&expo

Featured technologies and topics will include:

- Interoperability
- Enterprise Networks
- Securing Web Services
- Integrating existing networks
- Leveraging your existing software
- Real Time Web Services
- Where and When Should I Use Web Services?
- Web Services: From Consumption to Publication
- UDDI

Register By February 14, 2003 SAVE $400

Register By March 14 SAVE $200

Hynes Convention Center
Boston, MA

Boston
March 18-20

London
June 3-5

Berlin
June 24-26

Hong Kong
Coming soon...

For more information visit
www.sys-con.com

Over 200 participating companies will display and demonstrate over 500 developer products and solutions.

Over 3,000 Systems Integrators, System Architects, Developers and Project Managers will attend the conference expo.

Over 100 of the latest sessions on training, certifications, seminars, case-studies, and panel discussions promise to deliver REAL Web Services Benefits, the industry pulse and proven strategies.

Contact information: U.S. Events: 201 802-3069 or e-mail grisha@sys-con.com • European & Asian Events: 011 44 208 232 1600

OWNED BY SYS-CON MEDIA
PRODUCED BY SYS-CON EVENTS

Conference program available online!
www.sys-con.com/webservices2003east

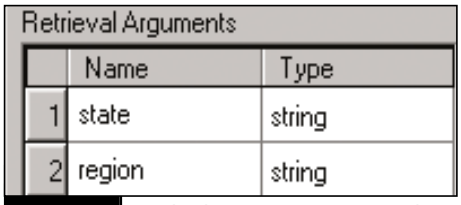


FIGURE 5 Retrieval arguments as a general property

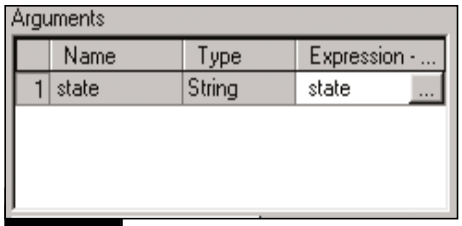


FIGURE 6 Supplying retrieval arguments to a nested report

DataWindow or one of a different presentation style, the nested report does not need to exist at design time; it can be created dynamically. The following code example dynamically creates a nested report:

```
String    ls_modstring
String    ls_rc

ls_modstring =     'create report(band=sum-
mary dataobject="d_composite_tab" ' + &
  'x="1083" y="12" height="796"
width="1001" border="0" ' + &
  'height.autosize=yes criteria=""
trail_footer = yes  name=dw_nested_2 ' + &
  'visible="1"  slideup=directlyabove)'
dw_1.Modify(ls_modstring)
```

The nested report named dw_nested_2 uses d_composite_tab as its DataWindow object. It's important to point out that even though nested reports can be created dynamically, the DataWindow objects that they are associated with cannot. In this example, d_composite_tab must already exist. It can't be created dynamically via a SyntaxFromSQL() function. Although SyntaxFromSQL can be used to dynamically create a DataWindow object to be used as the source of a DataWindow or DataStore, it's no help to us when using nested reports.

Sharing Data Between Reports

Because nested reports within a composite DataWindow are truly DataWindow children, the same result set can be shared among them with a ShareData() method. Of course, the column sets must match in all of the nested DataWindows that share the result set. Sharing a result set is much faster than retrieving all nested reports individually as only one database retrieve needs to be performed. The following code retrieves a result set into a DataStore, then shares it with two nested reports:

```
DataStore                 lds_prod_info
DataWindowChild           ldwc_nest1
DataWindowChild           ldwc_nest2
lds_prod_info = CREATE DataStore
lds_prod_info.DataObject =
"d_composite_tab"
lds_prod_info.SetTransObject(SQLCA)
lds_prod_info.Retrieve()
dw_1.GetChild("dw_nested_1",ldwc_nest1)
dw_1.GetChild("dw_nested_2",ldwc_nest2)
lds_prod_info.ShareData(ldwc_nest1)
lds_prod_info.ShareData(ldwc_nest2)
```

Notice the composite DataWindow does not require a transaction object or a database retrieve. The DataStore handles database transactions. Remember that each time data is retrieved into the composite DataWindow, all references to DataWindow children are lost. Always make sure to reestablish the child handle with a new call to the GetChild() function before trying to share data.

LIMITATIONS ON SHARING DATA

Using the ShareData() function produces some unexpected results on nested reports. When you use ShareData() with nested DataWindows that are part of a composite DataWindow, there are some instances where the ShareData will not work. Use the following rules of thumb:

- Sharing data from a DataWindow or a DataStore with a nested report – okay
- Sharing data from a nested report with a DataWindow or DataStore – will not work
- Sharing data from a nested report with another nested report – will not work

These three rules apply to all nested reports, regardless of whether or not they are being used in a composite DataWindow presentation style. A good way to get around this problem is to use RowsCopy when the ShareData() function will not work. An example of the RowsCopy() function is:

```
dw_1.RowsCopy (1, 1, primary!, dw_2, 1,
primary! )
```

Of course, whenever you call a ShareData() or RowsCopy() function, both the source and destination column sets must match.

Limitations Nesting/Composite

Usually nested reports tend to be of the more common DataWindow presentation styles such as FreeForm, Tabular, or Grid. For the most part you're allowed to nest the various types of DataWindow presentation styles. However, significant limitations apply to two of them.

- ***Crosstab:*** You can't place a crosstab with retrieval arguments within another report as a related nested report. However, you can include a crosstab in a composite report.
- ***RichText:*** RichText reports cannot be nested in any way. You are not allowed to place a RichText report in another report and you cannot place a RichText report within a composite DataWindow.

What's the Difference? Composite vs Nested

Nested reports are not a DataWindow style, but DataWindow objects used within DataWindows. Nested reports are usually hierarchical – they have a detail relationship with the base DataWindow. To create a nested report, open the base DataWindow, then place a report object in the desired DataWindow band. The nested report can be linked to the base DataWindow by setting its retrieval argument property. The criteria for nested report retrieval can be a column value (in the base DataWindow) or a constant.

When retrieving a base DataWindow containing a nested report, the base DataWindow is retrieved first, then each of the nested reports. In most circumstances, nested re-reports are not considered DataWindow children, therefore they cannot be accessed via a GetChild() method. There is an unsupported workaround that will allow the programmer to treat the nested report as a true DataWindow child.

> "Nested reports are usually hierarchical – they have a detail relationship with the base DataWindow"

Composite reports are a DataWindow presentation style that provides the programmer with a place to put unrelated mutually independent DataWindow objects. When run, each DataWindow is retrieved independently; no relationship exists with either the base DataWindow or any other DataWindow within the composite report presentation style. When supplying retrieval arguments, they must be supplied in the code.

Unlike using nested reports, the composite report presentation style treats the contained DataWindow objects as DataWindow children so the GetChild() method can be used to access the report and any objected contained within it. Just remember to name the report objects in the composite DataWindow if you want to obtain its handle via a GetChild() method. Similar to using nested reports, the composite report DataWindow presentation style is used when the data is read only. When the user runs a DataWindow that either contains nested reports or was built with the composite report presentation style, the base DataWindow is rendered in Print Preview mode. Both composite and nested reports are true reports and are, therefore, never updateable. ▼

Author Bio

Bob Hendry is a PowerBuilder instructor for Envision Software Systems and a frequent speaker at national and international PowerBuilder conferences. He specializes in PFC development and has written two books on the subject, including Programming with the PFC 6.0.

bob@sys-con.com

WebServices JOURNAL
.NET J2EE XML

The Best .NET Coverage Guaranteed!

LEARN WEB SERVICES. GET A NEW JOB *!*

Subscribe today to the world's leading Web Services resource

Get Up to Speed with the Fourth Wave in Software Development

- **Real-World Web Services**: XML's Killer App*!*
- How to Use **SOAP** in the Enterprise
- Demystifying **ebXML** for success
- **Authentication**, **Authorization**, and **Auditing**
- **BPM** - Business Process Management
- Latest Information on **Evolving Standards**
- Vital technology **insights** from the nation's leading Technologists
- Industry **Case Studies** and **Success Stories**
- Making the Most of **.NET**
- **Web Services Security**
- How to Develop and Market Your Web Services
- **EAI** and Application Integration Tips
- **The Marketplace**: Tools, Engines, and Servers
- Integrating **XML** in a Web Services Environment
- **Wireless**: Enable Your **WAP** Projects and Build **Wireless Applications** with Web Services*!*
- Real-World **UDDI**
- **Swing**-Compliant Web Services
- *and much, much more!*

Only $69.99 for 1 year (12 issues)*

* Newsstand price $83.88 for 1 year

Subscribe online at www.wsj2.com or call 888 303-5252

*Offer subject to change without notice

EVALUATING and ADOPTING WEB SERVICES
A pragmatist's view
UNDERSTANDING THE WEB SERVICES VISION
Scalable Web Services Using JMS & JCACHE
WebServices JOURNAL
.NET or J2EE

SYS-CON MEDIA

SYS-CON Media, the world's leading *i*-technology publisher of developer magazines and journals, brings you the most comprehensive coverage of Web services.

Dynamic DataWindow Restoration and Retrieval

WRITTEN BY IGOR SKOKIN AND CHARLES SILBERGLEITH

Add reports to existing applications without recompiling

The task of restoring the dataobject of a DataWindow (DW) or DataStore (DS) dynamically at runtime has been discussed many times in *PBDJ*.

One method is to use DW syntax, but it has some limitations. It doesn't work for composite DWs or DWs with nested reports. If a DW contains DropDownDataWindows (DDDW), they're not restored automatically. An elegant way of restoring them is shown in the article "Dynamically Creating or Modifying DropDownDataWindows (DDDWs) from Syntax" at http://my.sybase.com/detail?id=1001992.

Another problem is how to build generic code to retrieve a DW with retrieval arguments. The article "Report Deployment Using Generic DataWindow Retrieval" by Sameer S. Kulkarni (*PBDJ*, Vol. 7, issue 4) describes a service that replaces references to retrieval arguments in DW syntax with their actual values (the service doesn't process lists of values).

In this article we show how to use full state to restore DWs dynamically and how to use temporary tables to set values or lists of values for DW retrieval arguments. The method allows restoring DWs with DDDWs, nested reports, and composite DWs. It may be used to add reports to an existing application without having to recompile it.

Restore and Retrieve Algorithms for a Report/Criteria DataWindow

For the sake of brevity, let's consider restoring a report and criteria DWs. The criteria DW lets the user set retrieval arguments; the report DW shows the report itself. Both DW controls are based on the standard visual user object "u_dw_base"; when a DataStore is to be used it's based on the standard class user object "u_ds_base". To restore a DW by its full state stored in a variable of type BLOB, simply use the function SetFullstate() call. How to prepare full state to store in a database and how to retrieve data in a restored DW depends on the DW type. We'll consider the following DW types: a plain DW with or without DDDWs, a composite DW, and a DW with nested reports.

To store full state the following table was created:

```
create table AR_FULLSTATE (
REPORT_ID numeric(8) not null,
DATAOBJECT varchar(100) not null,
 FULL_STATE text null)
```

The column REPORT_ID identifies a report to be processed, the DATAOBJECT specifies the dataobject of a DW to be restored, and the FULL_STATE contains corresponding full state. This table is used to restore report and criteria DWs and if needed all nested DWs. Both user objects u_dw_base and u_ds_base have the function of_set_dataobject() that takes two parameters: long al_report_id is the report ID, and string as_dataobect is the dataobject name (see Listing 1).

Listing 2 provides the text of the function that stores a BLOB in the database. It takes three arguments: report ID, the dataobject, and the corresponding full state.

Restore and Retrieve Algorithms for a Plain DataWindow with or Without DDDWs

A plain DW may be restored by its syntax, but DDDWs will not be restored automatically. With the use of the full state, all DDDWs are restored automatically as long as they're placed into the detail band of the parent DW. If a DW has DDDWs in bands other than details, they have to be replaced with computed fields based on the LookUpDisplay() function call for corresponding DDDWs, and DDDWs themselves have to be moved into the detail band and made invisible.

To obtain a DW full state that allows its restoration at runtime, the code has to assign its dataobject to a DW or DS and call the GetFullState() function. No additional processing is required. The retrieve of such a DW is straightforward once all the retrieval arguments are set in temporary table(s): by the Retrieve() function call without arguments.

Restore and Retrieve Algorithms for a Composite DataWindow

It's possible to re-create all the nested reports of a composite DW through full state if all the nested reports are plain DWs. They can't contain nested reports or DDDWs; all of them will be lost after a composite DW is restored.

The code to save data for a composite DW so that it may be restored at runtime is in Listing 3. The variable adw references a DW control with the composite DW; no retrieves are done. The algorithm goes through each nested report and inserts a row in it, then the full state of such a DW may be saved. Also all the nested reports' full states are saved in the database. The code to retrieve the restored report is in Listing 4. (Listings 4–15 can be downloaded from www.sys-con.com/pbdj/sourcec.cfm.) A simple Retrieve() call for the restored DW won't work. To solve the problem we have to populate all nested reports in a loop. At each loop step, the code gets the dataobject property for the current nested report and restores it in a variable "ds" of type u_ds_base. Then the Retrieve() function call without retrieval arguments populates the DataStore ds and the data

is copied into the corresponding nested report with the RowsCopy() function.

Restore and Retrieve Algorithms for a DataWindow with Nested Reports

To process a DW of this type we need references to all nested reports (NRs). The function GetChild() may not be used with NRs, so we have to use the dot notation. This means that the names of all NRs have to be hard coded. Here we use the names from "nested_1" to "nested_20" for NRs.

To address NRs we need a function that returns the dwobject reference to an NR by its name and row number. We need one such function for DWs and one for DSs. Listing 5 shows the script of both functions. It takes three arguments: a reference to a parent DW or DS, an NR name, and a row number.

The code to save data for a DW with NRs so that it may be restored at runtime is in Listing 6. The variable adw references a DW control with the report, no retrieves done. The code inserts a row in the DW, then loops through NRs and inserts a row in each. Because the function InsertRow() may not be called for NRs, a DS with the same dataobject is created, a row is inserted there, and the data is copied into the NR using the dot notation. The full state of such a DW may be saved. Also all the NRs' full states are saved in the database.

To retrieve the restored report see Listing 7. First the parent DW is retrieved with the Retrieve() function call without arguments. Then in a loop the code checks whether the DW has NRs named from "nested_1" to "nested_20". All found NRs are populated with data in each row. For this an object named "ds" of u_ds_base type is created, the nested report's DW is restored in it using the NR's dataobject property, the code retrieves data into "ds", then the data is copied into the corresponding NR using the dot notation. The retrieval arguments are obtained using the Evaluate() function for the corresponding row of the parent DW.

Retrieval arguments of each NR are described in both the parent DW and the NR. Using the former, the code gets actual values; the latter allows you to replace references to the NR's retrieval arguments with the actual values. Then retrieve for the DataStore "ds" may be done with the Retrieve() function call without arguments. To avoid a retrieval arguments popup window at retrieve, the code removes the description of the retrieval arguments from the syntax of the NR. Listing 8 shows the code of the function of_get_row_select() that does the trick. The function takes the following arguments: adw is the reference to the DW control, al_row is the current row number, as_select is the SELECT statement of the NR, as_args_main is the parent DW's description of the NR's retrieval arguments, and as_args_nested is the description of the retrieval arguments in the NR. To obtain the parent DW's description of the NR's retrieval arguments, the Describe() function is used as follows:

```
Describe("{nest report name}"+&
".nest_arguments")
```

If, for example, an NR has three retrieval arguments corresponding to the following columns of the parent DW – "sval" of string type, "lval" of numeric type, and "dtval" of datetime type – the returned expression is (("sval"),("lval"),("dtval")). If a retrieval argument is an expression, it will be placed within quotations, like ("lval+2"). After all the expressions have been evaluated, we have to take the SELECT expression of the NR and substitute all references to the retrieval arguments with actual values. The list of retrieval arguments of an NR may be obtained from its syntax. For the mentioned example, it looks like (("ra_string",string),("ra_long",number),("ra_datetime",datetime)), where names of retrieval arguments are put in quotations. The order of arguments in the expression given by the "nest_arguments" property matches the order of the elements of the retrieval arguments list.

To substitute all references to retrieval arguments in the SELECT expression with the actual values, the function f_global_replace() is used (see Listing 9).

Processing of the Criteria Information

The criteria DW allows for the setting of values (or lists of values) as arguments to the report DW retrieval. All values are passed to the report DataWindow using temporary tables. To set one value we may use either a DDDW or an edit/editmask field. To set a list of values we'll open a browser window with a list of available values for the column and a checkbox against each value. The code of the report retrieval will create and populate all necessary temporary tables. Below all columns are considered independent: values set in one don't interfere with values set in all other columns. The values entered in columns will not be validated by the code. Also the code will not enforce the "required" property of columns.

The information retrieved from the database parameters about the criteria DW as well as about criteria values set by a user is stored in the window's instance array – one element per column of the criteria DW. The array also has info on the temporary tables to pass criteria values to the retrieving SELECT or stored procedure. The type of the array is a structure named str_column_info that contains the following fields:

- ***column_name – string:*** The name of the column in the criteria DW.
- ***temptable_syntax – string:*** The syntax of a temp table corresponding to the column.
- ***temptable_name – string:*** The name of the temporary table extracted from temptable_syntax.
- ***ds_temptable_link – DS:*** The link to a temporary table, its DataWindow is created dynamically and contains all temp table's columns. To insert criteria data into a temp table the code inserts a row(s) in this DS and calls the Update() function.
- ***ds_interface – DS:*** If a browser window will be used to set a list of values for the column, this DS will store the presentation of the browser DW. When the browser window is closed, this DS will keep information about checkboxes selected by a user.

The code for the function of_prepare_parameters() to initialize the array is in Listing 10. The data retrieved from the database is stored in the array named "i_str_colinfo". In the database, the information is stored in the table AR_PARAMETERS with the following syntax:

```
create table AR_PARAMETERS (
REPORT_ID numeric(8) not null,
COLUMN_NAME varchar(100) not null,
DW_SYNTAX text null,
TABLE_SYNTAX text null)
```

The column DW_SYNTAX contains syntax for the browser DW (if required) to be used to create the DS ds_interface. The column TABLE_SYNTAX describes syntax to create a temporary table. The function takes one argument: the criteria DW control. The instance variable i_l_report_id is equal to the ID of the current report. The code goes through the DW's columns and retrieves corresponding values of the columns DW_SYNTAX and TABLE_SYNTAX for each one. If the DW_SYNTAX value isn't empty, the DS ds_interface is dynamically created. Then the code creates the temp table using the TABLE_SYNTAX value and creates the DS ds_temptable_link dynamically with the source "Select * from {created temporary table}". Finally the DS is modified to become updateable.

The DS ds_interface is used if a browser window is required. The browser window is opened with a parameter of array i_str_colinfo[] element for the corresponding column. The structure is stored in the browser's instance variable i_str_column. The browser window has one DW object dw_choices for the user to select values out of the list. It also has OK and Cancel buttons. In its open event the window populates the dw_choices control with what we have in the DS ds_interface:

```
blob blb_state
i_str_column.ds_interface.&
GetFullState(blb_state)
dw_choices.SetFullState(blb_state)
```

The code for the OK button updates the DS ds_temptable_link with users' choices (see Listing 11). The DW must have the column "selected" as a checkbox, and the value one corresponds to "checked". Also the column name for the processing of dw_choices is expected to be the same as the variable i_str_column.column_name. We expect that

the first column of the temporary table is the column to be used as the retrieval criteria – thus the code refers to column number one to populate the DS ds_temptable_link.

In case a column does not require a browser but just one value, we use the itemchanged event of the criteria DW. The event's code calls the function of_modify_paraminfo(), passing it all three parameters of the itemchanged event. The code of the function is in Listing 12, and is very similar to the code of the OK button in Listing 11.

Finally, the function of_prepare_criteria() prepares all temporary tables before a report retrieval (see Listing 13). It drops and re-creates all temporary tables – in case another report window modified them for itself – and then calls the Update() function for the ds_temptable_link DataStore of each element of the i_str_colinfo array. The second argument of the function is set to False, not to reset flags, thus allowing temp table INSERTs the next time the function of_prepare_criteria() is called. The report retrieve follows immediately.

Let's look at an example of a report's stored procedure that uses a temporary table. Assume we have a table that describes stocks and we want to retrieve information about certain ones. If a user does not select anything, we want all stocks to be included in the report. Assume the information is stored in the table STOCK with the integer field STOCK_ID as a primary key. The temporary table populated with IDs of stocks selected by the user has the following syntax:

```
CREATE TABLE
#t_stock(id int not null)
```

In this case the stored procedure to show info on stocks in this way may look like Listing 14.

Maintenance Window – Main Elements

Let's look at two maintenance windows: one to store the info on report DW and another to store the info on criteria DWs and their columns. Both windows are inherited from a base window that has a DW control named dw_datawindows. It presents a user with a list of all DWs contained within a selected PBL. To get the list, the function LibraryDirectory() is used. It takes two arguments: the path to a PBL and the enumerated value DirDataWindow! to specify we need only the list of DataWindows. The function returns a tab-separated string of DW names, dates of modification, and comments. The string may be imported into the dw_datawindows control with the ImportString() function. Its dataobject is an external DW with the first three columns set up to accommodate the string returned by the LibraryDirectory() function: their types are string, datetime, and string. Other columns required for processing may be appended. In this case we need the ID of a report where a selected DW is used as either a criteria or a report. The code to populate the DW d_datawindows is in Listing 15.

After a user sets the report ID for a DataWindow, it may be stored in the database using the code for the DW's type: plain, composite (see Listing 3), or with nested reports (see Listing 6). The window that describes criteria DWs needs an additional DW control in which a user specifies temporary table syntax to pass the criteria value or a list of values for each column. Also the dataobject for the browser window is set if needed.

Conclusion

The described algorithm allows you to add new reports to an existing application without redeploying it. It may be further enhanced by adding links between criteria columns (when a value set in one of them limits a list of available values for the others), processing criteria other than list of values (for example, the LIKE criteria), and more.

Acknowledgements

We would like to thank Angelo Delgiudice and Daniel McNatty. Without their help this article would never have been completed. ▼

Author Bios

Igor Skokin is an assistant vice president for Deutsche Bank in Manhattan. Igor holds a master's degree in theoretical physics from Kiev State University in the Ukraine.

Charles Silbergleith is an independent consultant and has been working in the financial industry for over 20 years building a variety of applications from mainframes to Web sites.

igor.skokin@db.com

csilbergleith@earthlink.net

Listing 1 Restoring a DataWindow or DataStore by its full state

```
public subroutine of_set_dataobject(long al_report_id,
string as_dataobject)
blob blb_state
SELECTBLOB FULL_STATE INTO :blb_state
FROM AR_FULLSTATE
WHERE REPORT_ID=:al_report_id AND
DATAOBJECT=:as_dataobect;
this.SetFullState(blb_state)
```

Listing 2 Storing a full-state BLOB in the database

```
public subroutine of_store_fullstate(long al_report_id,
string as_dataobject, blob ablb_state)
int i_count
SELECT count(*) INTO :i_count
FROM AR_FULLSTATE
WHERE REPORT_ID=:al_report_id AND DATAOBJECT=:as_dataob-
ject;
IF i_count=0 THEN
  INSERT INTO AR_FULLSTATE (REPORT_ID, DATAOBJECT)
  VALUES(:al_report_id,:as_dataobject);
END IF
UPDATEBLOB AR_FULLSTATE SET FULL_STATE=:ablb_state
WHERE REPORT_ID=:al_report_id AND DATAOBJECT=:as_dataob-
ject;
```

Listing 3 Composite DataWindow: saving the full state

```
// adw is a reference to the DW to process
int i
string s_objects,s_current_obj
blob blb_state
datawindowchild dwc
datastore ds
IF adw.Describe("DataWindow.Processing")<>"5" THEN Return
ds = create datastore
s_objects = adw.Describe("datawindow.objects")+"~t"
DO WHILE Pos(s_objects, "~t") > 0
  s_current_obj = Left(s_objects, Pos(s_objects, "~t") - 1)
  s_objects = Mid(s_objects, Pos(s_objects, "~t") + 1)
  IF adw.Describe(s_current_obj+".type") = "report" THEN
    IF adw.GetChild(s_current_obj, dwc) = 1 THEN
      dwc.InsertRow(0)
      ds.dataobject=adw.Describe(s_current_obj+".dataob-
ject")
      ds.GetFullState(blb_state)

of_store_fullstate(il_report_id,ds.dataobject,blb_state)
    END IF
  END IF
LOOP
adw.GetFullState(blb_state)
of_store_fullstate(il_report_id,adw.dataobject,blb_state)
```

PB9 Books

www.pb9books.com

DDDW Tips and Tricks

One step closer to PowerBuilder expertise

WRITTEN BY
BERNDT HAMBOECK

The DropDownDataWindow (DDDW) edit style is one of PowerBuilder's outstanding features. Yes, I know there are a lot of new and exciting capabilities in the upcoming release of PowerBuilder, but in this article I'll try to solve some of the current problems with the existing features that are popping up in nearly every project I've seen.

Here I'll focus on DropDownDataWindows, including:

- How to get started with DDDWs
- Filtering DDDWs without losing the display value in other rows
- Catching the collapsing of a DDDW
- Trapping the cursor keys in a DDDW
- Autocomplete DDDW values

Getting Started

The basics are well covered in the PowerBuilder User's Manual, but if you still have problems defining them look at Figure 1. Click on the column you want and its edit style (yes, click on edit on the properties) as DDDW (choose DropDownDW as Style Type).

Next, choose your DataWindow, display column, and data column. The display column is the kind of data that's displayed to the user. The data column value is the one that's saved into the database.

You might also consider checking the "V ScrollBar" property. This is something a lot of users forget; when you test your application, you have just a few rows in your DDDW. When you go into production you'll have a lot of rows there, but now the user can't scroll with the mouse to these rows.

Now we'll start by describing some problems beginners frequently run into when they are using DDDWs.

The first problem I've encountered as a trainer in the Fast Track to PowerBuilder classes is the following:

A student creates a DataWindow and wants to define a column with the DropDownDataWindow edit style. He or she opens a second DataWindow painter and creates the DDDW and saves it (remember the naming convention for DDDWs? Yes, it's d_dddw_xxxx, where xxxx is your part of the name). Now the student switches back to the column and configures the edit style. The next step is to test the DDDW column by inserting a new row or collapsing the column on an existing row. Well, the student expects to see his or her DDDW data from the database, but that's wrong. The DDDW is empty (see Figure 2), but why? Well, the DDDW is filled only after a retrieve of the DataWindow. So the student reretrieves the data and voilà, everything is fine when we look at the column.

The next thing we should talk about is DDDWs with a retrieval argument. I'm quite sure you've already seen that: if you use a DDDW that has retrieval arguments, PowerBuilder will prompt the user for those arguments, displaying an ugly window when you retrieve the parent DataWindow. This is, of course, not the normal behavior you want your application to have.

The solution is to retrieve the DDDW before you get data for the primary DataWindow. The problem is that we need a reference to the DDDW so we can issue a Retrieve() or InsertRow() function on it. To get the object handle to the DDDW, use the GetChild() function.

Since PowerBuilder 8 we are able to prevent the retrieve of the DDDW by disabling the "Autoretrieve" property. This lets us retrieve the DDDW anytime (also after the retrieve of the master DataWindow).

We can also prevent the retrieval of a DDDW by saving it with a blank row and adding a row on the DDDW in the DataWindow painter. To do so click on the column specification window on the data tab and insert a blank row using the right-mouse button. This saves the DataWindow with one blank row preloaded for us. Since PowerBuilder will see at least one row in the DDDW, it won't try to issue its own retrieve. This also allows us to retrieve the DDDW anytime.

Now that we know how to prevent the retrieve of a DDDW, the next step is to get the DDDW handle and retrieve it alone. We can accomplish this by using the GetChild function. Normally when you interact with a column on a DataWindow, you use the column name to get or set the current value. However, we want to interact with the DDDW that is on a column, not with the column itself. GetChild() provides us with a reference variable that points to the actual DataWindow in memory. We can use this variable to issue functions against the DDDW such as Retrieve(), InsertRow(), and Modify().

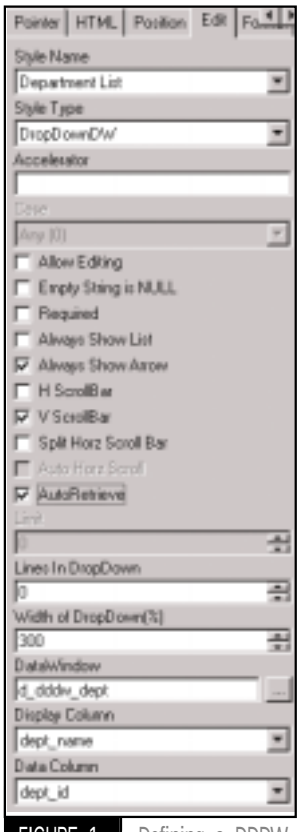


FIGURE 1 Defining a DDDW

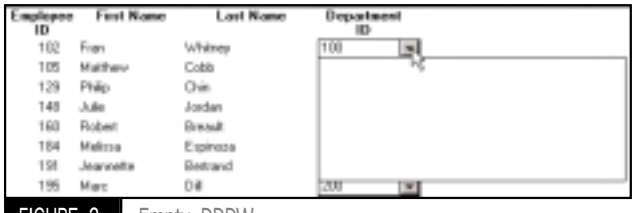


FIGURE 2 Empty DDDW

Here's a sample of how to use GetChild():

```
DataWindowChild dwc
IF dw_1.GetChild( "dept_id", dwc ) >
0 THEN
   dwc.SetTransObject( SQLCA )
   IF dwc.Retrieve() = 0 THEN &
      dwc.InsertRow(0)
END IF
```

Notice that GetChild returns an integer. This integer tells you whether GetChild was able to return a reference to the child DataWindow into the variable dwc. GetChild looks on your parent DataWindow for the column you specify. It then assigns the DataWindow control used to retrieve that column to the variable dwc.

If the column "dept_id" is a DDDW, then dwc should hold a reference to the DDDW. If we had misspelled the name of the column or the column didn't exist on the DataWindow, GetChild() would return -1. As usual don't forget to use SetTransObject on the child before you code a retrieve. Once we have our reference to the dropdown, we can do almost anything with it that we can do to a normal DataWindow including using Modify, Describe, SetTransObject, Find, Sort, and Filter.

Be sure that the data value for a column that uses the DropDownDataWindow edit style is limited to 511 characters.

Share DropDownDataWindows

If you use a lot of DDDWs, it might be a good idea to populate a DataStore for each DDDW and share with the DataStore. An important point to remember is that you *must* prevent the retrieve of the DDDWs prior to sharing (this is usually accomplished in the open event of the window) or the DDDWs will initially do a retrieve. If this happens, you will be retrieving twice for each DDDW! To stop the retrieve of the DDDW, do an insert row into the ChildDataWindow from the open event of the window.

Get the Display Value

I've come across a problem with a PB 6.5 app: on a DataWindow I have a DDDW that accesses a simple two-column (name, number) table. The DDDW uses "name" as its display column and "number" as its data column. The user is able to select a name from the list, and the number is stored in another table. So far, so good.

The problem occurs when I want to programmatically access the data contained in the display column of the DDDW (i.e., the name the user selected). I have gone through all the documentation I can find, and I can't figure out how to obtain this information.

SOLUTION #1 (RECOMMENDED)

Take advantage of the Describe() Evaluate function. In the following code the DDLB or DDDW column is called state_code.

```
string ls_rownumber, ls_displayvalue

ls_rownumber = string(dw_1.getrow())
ls_displayvalue =
dw_1.describe("Evaluate( 'lookupdis-
play(dept_id) ', "+ls_rownumber+" )")
```

This solution does not require the definition of an additional computed column on the DataWindow.

Note: This solution will not work in the itemchanged event of the main DataWindow. It must be done in an event that occurs after the itemchanged event has completed by creating a custom unmapped user event ue_ getdisplayvalue, and then windowname.-postevent (ue_getdisplayvalue) from the itemchanged event of the DataWindow.

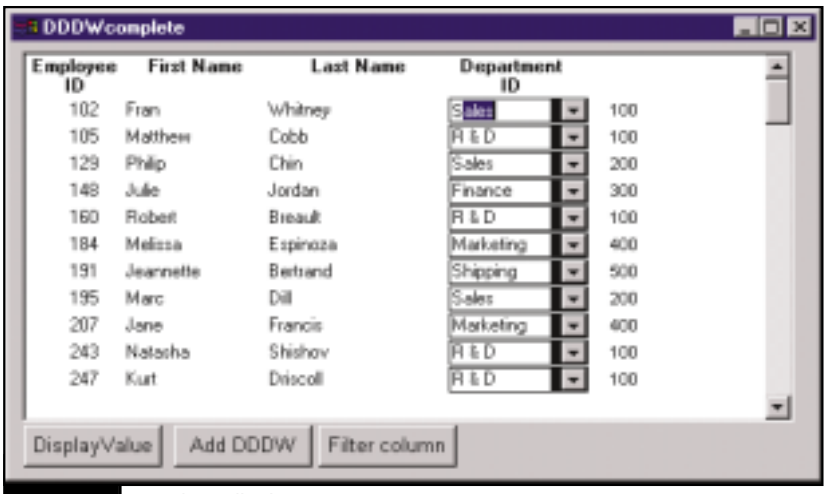

DDDWcomplete
Employee ID
First Name
Last Name
Department ID
102 Fran Whitney Sales 100
105 Matthew Cobb R & D 100
129 Philip Chin Sales 200
148 Julie Jordan Finance 300
160 Robert Breault R & D 100
184 Melissa Espinoza Marketing 400
191 Jeannette Bertrand Shipping 500
195 Marc Dill Sales 200
207 Jane Francis Marketing 400
243 Natasha Shishov R & D 100
247 Kurt Driscoll R & D 100
DisplayValue
Add DDDW
Filter column

FIGURE 3 Sample application

SOLUTION #2 (COMPUTED COLUMN APPROACH)

Go into the DataWindow painter and add a computed column. The expression should be Lookupdisplay(dept_id) where dept_id is the name of your DDLB or DDDW column. Name the computed column "display". Place the computed column anywhere since we will make it invisible with Modify().

Next, go into the window and add a user event called ue_lookup. In the script for this event code:

```
// displayvalue will contain the dis-
play value the user has selected from
ddlb or dddw column
string displayvalue
displayvalue =
dw_1.getitemstring(dw_1.getrow(),"dis-
play")
```

In the Itemchanged event for the main DataWindow (dw_1) code:

```
// check to see if the ddlb or dddw
column is the correct one they are
changing.
// use the column number (#) of the
ddlb or dddw column.
if getcolumn() = 3 then
  parent.event post ue_lookup()
end if
```

ADDING A DDDW AT RUNTIME

I had some trouble getting this to work; apparently you have to reconnect to the database for it to function properly.

Here is an example that builds the Modify string with the attributes (see Listing 1) (Listings 1–3 can be downloaded from www.sys-con.com/pbdj/sourcec.cfm.):

- Make "dept_id" the data column.
- Make "dept_name" the display column.
- Insert an arrow on the dropdown list.
- Allow editing in the text box.
- Insert a vertical scroll bar on the dropdown list.

Be sure that we are assigning the DataObject of the child DataWindow dynamically, which might end in an error message in the compiled code only.

The GetChild function returns -1 when we try to reference the DropDownDataWindow. This is because PowerBuilder compiles only "used" objects into an executable. If we assign a DataObject (or in our case a DDDW) programmatically using a string, PowerBuilder does not recognize this DataObject as used. The solution is that we assign a pbd out of the pbl containing that particular DataObject (in our case "dw_popup_table_list"), or we should include it in a .pbr file as:

```
pibblename.pbl(dw_popup_table_list)
```

FILTER DDDWS

You may have already encountered this problem: we would like to filter a DDDW depending on a value in the current row. If we filter out some values in the DDDW, the dropdown shows only description data for rows that are in the primary buffer. In all other rows we see the data value.

The trick here is not to filter. You need to have the description or display value in the primary buffer. One method is to use SetDetailHeight to mimic "filtering."

- Unfilter all rows and set detail height to "standard height".
- Filter out rows you want to display (yes, the good ones).
- SetDetailHeight to 0 for the rows in the primary buffer (this will make them invisible).
- Unfilter the DDDW.

You'll be left with all the rows so the display value can be found but the rows you don't want to see have a height of 0 (see Listing 2).

The Cursor Keys

We can code an event using pbm_dwnkey to get the keys pressed by a user on our DataWindow; however, if you type the up or down arrows while the cursor is positioned on a DDDW, the itemchanged event usually gets triggered, bypassing the pbm_dwnkey event. This might be bad in some situations; for example, we have problems trapping as these keys are the first and last row in our child DataWindow. The solution is to trap these in a user event mapped to pbm_command.

From the win32 helpfile (WM_COMMAND):

The WM_COMMAND message is sent when the user selects a command item from a menu, when a control sends a notification message to its parent window, or when an accelerator keystroke is translated.

This means that in PowerBuilder, whenever an event occurs in the DataWindowChild it sends a notification code to the parent DataWindow. This code can be intercepted in the user event mapped to a pbm_command DataWindow event. If we want to experiment to determine which events you can intercept, try adding the following code to ue_command mapped to pbm_command:

```
mle_status.text += "hwndchild = " +
String(hwndchild) + ", " + "childid =
" + String(childid) + ", " + "notifi-
cationcode = " + String(notification-
code) + "~r~n"
```

The code needed to intercept the dropdown arrow is:

```
int li_rc
li_rc = this.GetChild("dept_id",
ldwc_ddlb)
IF childid = Handle(ldwc_ddlb) THEN
   CHOOSE CASE notificationcode
            CASE  2048
                  Post Event
ue_DDLBRowFocusChanged()
   END CHOOSE
END IF
```

Some samples for notification code are shown in Table 1.

One caveat, the pbm_command event is invoked for just about everything that goes on in a DW that's not already captured by another event. In other words, it will be invoked a lot. To limit the number of times the code in this event is invoked, you can map another user event to the event ID pbm_dwnmousemove. Don't put any code in this event. This will cause all mouse move events to go to the other event instead of the one mapped to pbm_command.

NUMBER	EVENT
768	RetrieveStart
769	Retrieve End
1024	UpdateStart
1025	UpdateEnd
1280	DoubleClicked
1281	Clicked, RightDoubleClicked
1536	DBError
1792	ItemFocusChanged
2048	RowFocusChanged
2049	SelectionChanged
2311	Mouse Move
2313	LeftMouseButtonUp
2314	RightClicked
2315	RightButtonUp
2317	ScrollHorizontal
2318	ScrollVertical
2319	SQLPreview
2320	Resize
2321	PositionChanged

TABLE 1 Notification codes for WM_COMMAND

Collapsing the DDDW

This might also help you, if you have one or more rows in your DDDW where your display and data value appear more than once, but you want to set another column depending on a DDDW column that's not used for the display or data value. If you have such a row, PowerBuilder does not recognize (or fire) an itemchanged event, which (from PowerBuilder's point of view) is correct but does not solve our business problem. A good solution is to trap the cursor keys and look at which row the user is in in the DDDW and set the dependent column by hand.

You can achieve what you want by mapping one user event to pbm_dwndropdown to figure out when a DDDW was opened and another one to pbm_ncpaint, which is fired when a DataWindow needs to be repainted, and then code something like Listing 3.

Autocomplete DDDW

We would like to implement the type-along search functionality for a DropDownDataWindow. This means that when we start typing in our DDDW column we want the column to display the next matching value automatically (see Figure 3).

The steps to accomplish this are:

- Make your DDDW column editable.
- Code the editchanged (for DDLB) and itemfocuschanged (for DDDWs).

This can be found in the PFC. I've prepared a complete sample where I extracted the code from the PFC; it can be downloaded from *PBDJ*'s Web site, www.sys-con.com/pbdj/sourcec.cfm, or from www.bhitcon.net. In this sample you'll find the autocomplete feature as well as everything we were discussing.

Conclusion

The DropDownDataWindow (DDDW) edit style is one of PowerBuilder's most useful features. It's essentially a DataWindow within a DataWindow. Its most common use is as a listbox type control where the DataWindow that's actually dropped down looks and acts like a listbox control, allowing the user to select a row. But as we've seen, this is just the beginning of what can be done with dropdowns. DDDWs are a powerful feature in PowerBuilder; they're not easy to use, but if you know how to use them correctly you're one step closer to becoming a PowerBuilder expert. ▼

AUTHOR BIO

Berndt Hamboeck is a senior consultant for BHITCON. He's a CSI, SCAPC8, EASAC, and SCJP2 and started his Sybase development using PB5.

admin@bhitcon.net

ATTN: Developers

STEP UP to the mike and be...

Go to http://developer.sys-con.com

HEARD!

Calling Sleek Geeks Everywhere!

Make sure you have your finger on the pulse of i-Technology...bookmark http://developer.sys-con.com today.

i-Technology News

i-Technology Views

i-Technology Comment

i-Technology Debate

© COPYRIGHT 2002, SYS-CON MEDIA WWW.SYS-CON.COM

SYS-CON MEDIA

introductory subscription offer!

Here's what you'll find in every issue of .netdj:

Security Watch
Mobile .NET
.NET Trends
Tech Tips
Standards Watch
Business Alerts
.NET News
Book and Software Announcements

.NET Developer's Journal is for .NET developers of all levels, especially those "in the trenches" creating .NET code on a daily basis:

- For beginners: Each issue contains step-by-step tutorials.
- For intermediate developers: There are more advanced articles.
- For advanced .NET developers: In-depth technical articles and columns written by acknowledged .NET experts.

Regardless of their experience level, *.NET Developer's Journal* assumes that everyone reading it shares a common desire to understand as much about .NET – and the business forces shaping it – as possible. Our aim is to help bring our reader-developers closer and closer to that goal with each and every new issue!

SYS-CON MEDIA

SUBSCRIBE ONLINE!

www.sys-con.com/dotnet/

or Call

1 888 303-5282

SAVE 16% OFF

THE ANNUAL COVER PRICE

Get 12 issues of .NETDJ for only $69.99!

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

ANNUAL COVER PRICE:
~~$83.88~~
YOU PAY
$69.99
YOU SAVE
$13.89
OFF THE ANNUAL COVER PRICE

Wallpaper Setter in PowerBuilder

Reinvent the wheel with PB

WRITTEN BY
ATUL KANE

My son is only 18-months old but I have taken more pictures of him than I can count. Thanks to digital cameras it's so easy to take and store pictures.

Since I spend most of my time on the computer, I formed the habit of changing the Windows desktop with his latest portrait both at work and at home. However, the initial excitement faded due to my inherent laziness. I needed something automatic, some computer program to change the wallpaper for me. I also wanted something that would accept my new pictures without reconfiguring the program. I not only wanted a random effect, I also wanted the program to use all the pictures before it started repeating.

I searched the Internet and found several wallpaper-changing programs, but none of them did exactly what I wanted. Besides, I didn't want to take the risk of downloading some unknown 2MB file just to change the wallpaper either. I then decided to reinvent the wheel.

Can you guess which tool I used to build a program like this? Yes, you got it. PowerBuilder. It took me less than a day to write and test the wallpaper program using my favorite tool. Most people think of PowerBuilder as a traditional client/server tool. They're mistaken! Since it can call most Windows APIs, it also gives you the power to talk to the machine at a lower level and hence accomplish tasks for which there is no standard PowerBuilder function.

The rest of this article explains the important steps and techniques I used to create this little utility. Before I could begin the design, I wanted the Windows API that would change the wallpaper to the desired image file. I searched Microsoft's Knowledge Base and found the SystemParametersInfoA API, and gave it a shot. Although the Microsoft site used Windows 95 as an example, the API worked fine on my Windows 2000 machine.

Windows API Description

Let's declare this Local External Function on my Window.

```
Function Long SystemParametersInfoA (
uint uAction,  uint uParam,  String
lpvParam,  uint fuWinIni ) LIBRARY
"user32.dll"
```

Since this API does much more than deal with wallpapers, the first argument lets the API know what it's supposed to do. The second argument is always zero because the wallpaper setting does not require this argument. The third argument is the bitmap (BMP) filename, which is our wallpaper. The fourth argument is used to preserve the new settings. The purpose of this program is to set wallpapers and hundreds of them. Since I'll be constantly changing it, I'll set this argument to zero.

Given the above information, it's just a matter of one line of code to set the wallpaper from within PowerBuilder. Please note that the file has to be a bitmap file so I had to convert my JPG files to BMP.

I then quickly tested setting new wallpaper from within the PB application and started thinking about my other requirements.

Let's recap those – the program should be able to:

1. Set wallpapers one at a time from a given number of files
2. Perform Step 1 randomly
3. Not miss any file

For example, if the program knows about four pictures, a, b, c, and d, and this program is run four times and every time it's run it sets one of the four wallpapers, then the acceptable sequences are:

a, b, d, c
b, a, c, d
d, c, a, b etc.
a, b, a, d is not acceptable, though it's random, because it missed c.

What's the easiest way to tell the program about all the available files? I think most of you will agree that it is to create a folder and let the program display all the files from there. When new pictures are available, this folder can be updated accordingly. Also, one of the controls in PB can be used to populate all the filenames from a Windows folder again by one line of code. Yes, I'm talking about the good old Dirlist function for listbox.

At this point, to incorporate the desired randomness I can easily pick up one of the files from the folder using the rand() function, but that function alone is not enough because of my other requirements. Thus the displayed wallpaper information must be saved somewhere. A heavy-duty database is out of the question. A profile file function like SetProfileString, which is available in PB, immediately came to mind.

When the program runs for the first time, it creates the empty profile file with the required section. It then chooses a file randomly and in addition to setting the wallpaper, it saves its position in the listbox (tags recent1, recent2, etc.). It also saves the total number of wallpapers displayed in the profile file variable "lastrecent". This variable will be used to keep track of the number of times this program is run. When this number equals the number of files in the folder, all the wallpapers have been shown.

As long as the files in the directory remain the same (delete the profile file

whenever you add/remove picture files in the folder), the order of the files in the listbox is going to be the same no matter how many times the program is run, hence there's no need to save the filename. Also it's a whole lot easier to search if a particular wallpaper is already set. I'll build a string of all the previously shown wallpapers and see if the new random number is present. If it's not there in the list, I have found my new wallpaper. (I could have done this with a DataStore but I didn't want to create another object.)

When all the pictures in the directory are shown once, all the history in the profile file has to be cleared; re-creating the file does that as a side effect.

Before you build the executable change the instance variable as it stores the folder location where all your wallpapers are stored.

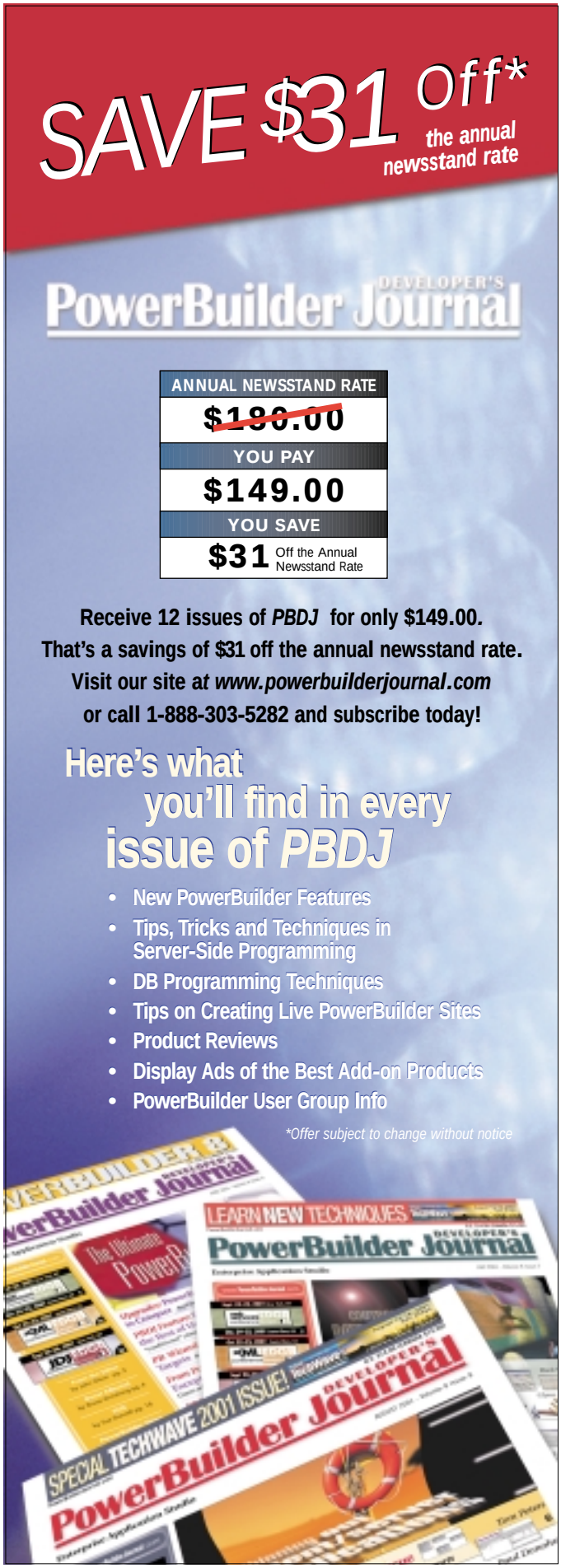
Now that the program is ready, you can use the Windows scheduler to run this program as often as you wish. I currently have over 200 pictures and the scheduler runs this program every 15 minutes. You can also run it as often as every minute to create a slide-show effect. If you prefer, you can always code the Windows timer event.

Note: If you like to run this program interactively, comment the close(this) in the open event of the window. ▼

akkane@yahoo.com

Author Bio

Atul Kane, an MIS manager for NEC USA, Inc., has worked with PB since version 3 and has 11 years of programming experience. He is also a Sybase-certified PowerBuilder developer associate, and a certified Sun Java programmer with a degree in computer science.


SAVE $31 Off*
the annual newsstand rate
PowerBuilder Developer's Journal
ANNUAL NEWSSTAND RATE
$180.00
YOU PAY
$149.00
YOU SAVE
$31 Off the Annual Newsstand Rate
Receive 12 issues of PBDJ for only $149.00.
That's a savings of $31 off the annual newsstand rate.
Visit our site at www.powerbuilderjournal.com
or call 1-888-303-5282 and subscribe today!
Here's what you'll find in every issue of PBDJ
• New PowerBuilder Features
• Tips, Tricks and Techniques in Server-Side Programming
• DB Programming Techniques
• Tips on Creating Live PowerBuilder Sites
• Product Reviews
• Display Ads of the Best Add-on Products
• PowerBuilder User Group Info
*Offer subject to change without notice
LEARN NEW TECHNIQUES
SPECIAL TECHWAVE 2001 ISSUE!

Mobile Development Tools

Gain a competitive edge

WRITTEN BY
SCOTT HEFFRON

recently presented a case study on mobile business at a local ColdFusion Users Group in Salt Lake City. Two questions dominated the Q&A section.

How productive can you be with a handheld and what tools can you use to create applications for it? To give you an idea of productivity, I'm writing the first part of this article at my daughter's swim meet. Trying to be a good father, I stop during her races to cheer her on.

As the workforce becomes more mobile, business people need to have customer information on hand in order to have a competitive edge. New devices and technologies, such as handhelds, Bluetooth, and Smartphones, are becoming available in the marketplace; businesses are trying to understand how to incorporate these devices into their infrastructure to get the advantage they desire. Not only do these devices have to be placed into the infrastructure, there also needs to be applications available in order for businesses to provide better service to their customers and get a full ROI for the company. The mobile configuration that I currently use is a Handspring Prism with the following accessories: folding keyboard, modem, digital camera, 16MB Flash memory, and scanner.

The mobile application world is fascinating. We need to create applications that can be distributed, so workers can do their jobs and not be tied to a desk. To improve worker performance, many companies want to put a focused amount of enterprise data into an application that resides in a pocket-size device. I think James Bond would be proud of what we can do these days. I'll focus on platforms and development tools that work with the UltraLite deployment tool.

Let's first look at handheld OS platforms. You'd think this would trim down the field on development tools, but not this time. The UltraLite can be deployed on the following platforms:

- Palm
- Windows CE
- Symbian

As you can see, the UltraLite deployment package works on major handheld platforms. This is nice to know, in case your company changes platforms because of feature requests or enterprise architectural changes. It doesn't happen often, but it can. Table 1 provides a matrix for UltraLite and supported platforms.

This brings us to which tools you can use to create handheld applications. There are several to choose from; it all depends on your criteria and what you want to accomplish. Review your requirements when looking at which tool you want to use. Realize also that there are implicit constraints on the hardware that you might be overlooking. These could include memory, speed, and screen size. Going from client/server or browser-based to handheld application development can take some getting used to. The following is a list of the tools that I reviewed that work with UltraLite.

- CodeWarrior
- MobileVB
- MobileBuilder
- Java
- Embedded Visual Basic

CodeWarrior from Metrowerks is for experienced "C" developers. It has a nice IDE and you can accomplish many things with it. One nice feature is the debug option with the emulator as it allows you to see how the variables and workflow are really being used. Its IDE history is from the Mac arena, so it may seem hard in the beginning, but it's easy to work through. I like this product. When I create an application for the Palm OS, I just send out one file, making it easy to deploy. There are tools for Palm, Symbian, and J2ME environments. One nice feature from Metrowerks is CodeWarriorU.com, an online tutorial section. Check it out; it has some good courses.

MobileVB from Appforge works on Palm, Pocket PC, and Symbian. If you understand Visual Basic, you won't have any problems with this tool. This is an add-on package for the VB Studio; it works on dragging and dropping visual objects onto the screen. You then code the events with business logic to complete the application. You'll need to purchase the Booster VM separately so the application can interpret the code effectively.

MobileBuilder from PenRight is a single custom application package designed to generate handheld applications specific to certain operating systems. This is a nice feature if you'll be jumping between different platforms. I don't see this happening too often, since many information system administrators don't like swapping devices; it causes headaches and heartburn. One thing about this software is that you also need a runtime library to operate your application on the device.

Java is used within the UltraLite Component Suite Toolkit. Native UltraLite for Java supports the Jeode VM on Windows CE/ARM devices including the Compaq iPAQ and NEC MobilePro P300, which have the Jeode VM. Windows operating systems other than Windows CE are supported for testing and development purposes only.

The *Embedded Visual Basic* piece is an ActiveX component targeted for 386 and ARM processor devices. I have not had a chance to do any research with this product, so I can't say much about it.

All these packages help create applications for handheld devices. You will need to know the available features in order to get a firm grasp on which package to use. Realize also that these pack-

UltraLite Version	Status (see below)	Windows CE Versions	Palm OS Versions	VxWorks Versions	JDK Versions	DOS Versions
8.0.2	Active	3.0 or higher	4.x or 3.x	5.4 or higher	1.1.4 or higher 1.3.x	not supported
8.0.1	Active	3.0 or higher, 2.11	4.x or 3.x	5.4 or higher	1.1.4 or higher 1.3.x	not supported
8.0.0	Inactive	3.0 or higher, 2.11	4.x or 3.x	5.4 or higher	1.1.4 or higher 1.3.x	not supported
7.0.4	Active	3.0 or higher, 2.11	2.0 or higher	5.3 or higher	1.1.4 or higher 1.2.x	6.02 or higher
7.0.3	Inactive	3.0, 2.11	2.0 or higher	5.3 or higher	1.1.4 or higher 1.2.x	6.02 or higher

TABLE 1 UltraLite and supported platforms

ages will leapfrog each other. This means that if one toolset has a feature that's liked by the general populace, the others will follow to make sure they're not losing clients. As these tools get more sophisticated, there will be libraries created to make development even easier. In many cases, these libraries already exist. You just have to know where to look and how to incorporate them.

- CodeWarrior: www.metrowerks.com
- MobileVB: www.appforge.com
- MobileBuilder: www.penright.com
- UltraLite Component Suite (Java, Embedded VB): www.sybase.com/developer/mobile/ulcs

As you can see many products are available that can be used to create applications for handheld devices. This list is just the beginning and there will be more coming down the pipeline as systems mature and developers push the limits of those systems. I use CodeWarrior for Palm development. I like the text-based and OO development approach. Reusing classes helps reduce development time. I also realize that each developer is different, so please check out what's available. I am looking forward to developing a J2ME application for a phone to see how it works.

As I was looking at the various targeted platforms, I noticed that the ARM processor is being used in Pocket PC systems. I'm looking forward to seeing what will happen with the Palm OS 5, since it's using the ARM processor. Will these products that are targeting the ARM processor also be able to use Palm OS devices along with Pocket PC devices? Only time will tell.

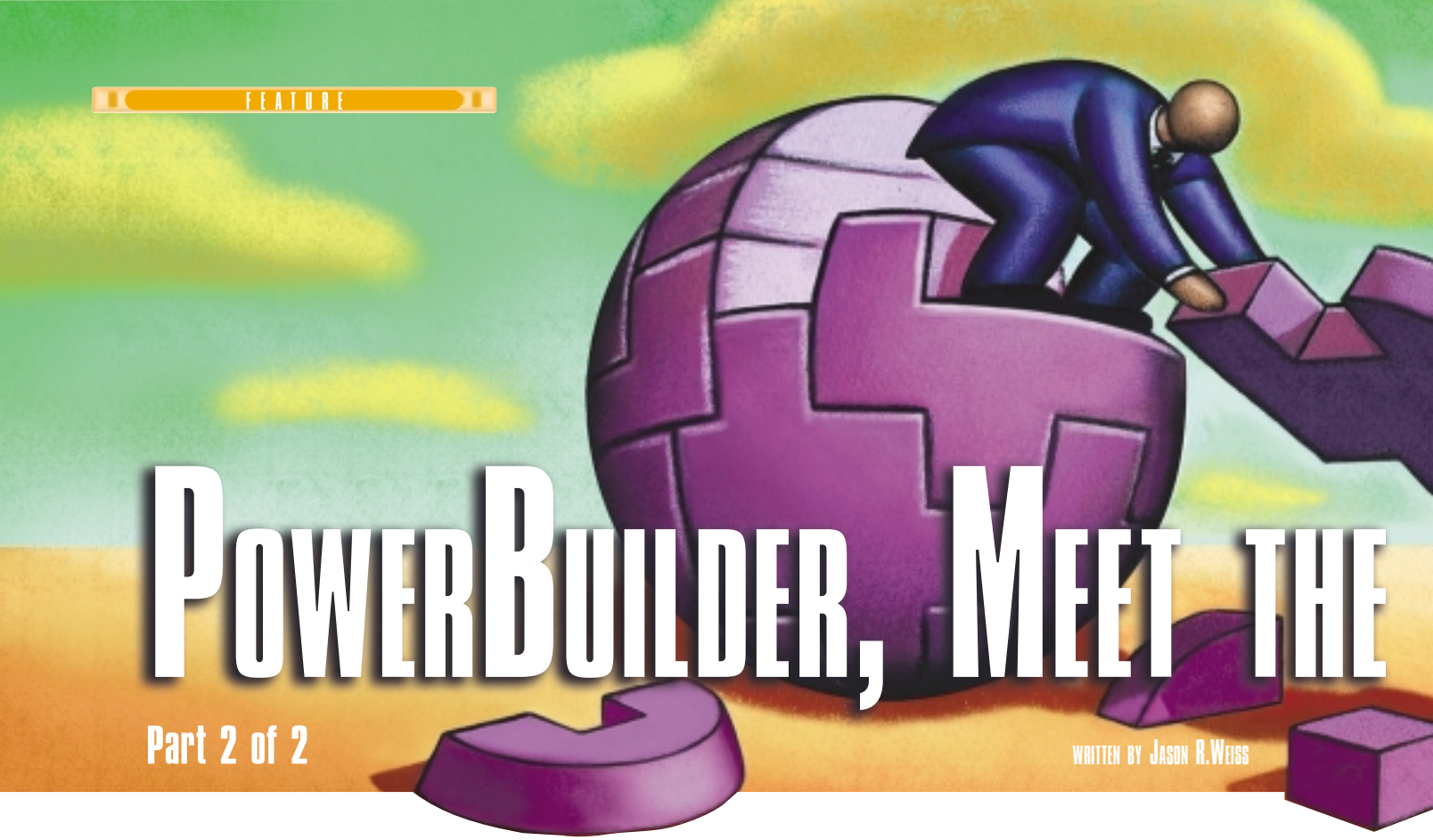
Businesses are starting to see how mobile applications can help them: a better quality of data is being entered into the system; the need for double entry or entering data from a fax by the salesperson has been removed; and customer service problems and requests are handled in a timely fashion. We are also seeing areas for better forecasting of products and services. This helps not only the company, but also the clients. It's a win/win solution for everyone. ▼

Author Bio

Scott Heffron is a senior software engineer in the Salt Lake City area. He has been involved in client/server, GIS, Web, and mobile computing projects for the past 16 years. He is currently working on handheld device technologies

heffrons@earthlink.net

The World's Leading Java Resource!
JAVA DEVELOPER'S JOURNAL
Here's what you'll find in every issue of JDJ:
Industry insights
The latest software trends
Technical expertise
Career opportunities
In-depth articles on Java technologies
Subscribe Today & SAVE 30% Off the annual cover price
ANNUAL COVER PRICE
$71.88
YOU PAY
$49.99
YOU SAVE
30% Off the annual cover price
Sign up ONLINE at www.javadevelopersjournal.com
OFFER SUBJECT TO CHANGE WITHOUT NOTICE
JDJEDGE CONFERENCE & EXPO



PowerBuilder, Meet the

Part 2 of 2

WRITTEN BY JASON R. WEISS

How can Sybase's EAServer architecture help create a bridge between the worlds of Java and PowerBuilder to access advanced cryptographic functionality that isn't natively available in PowerBuilder?

Part 1 of this article ("A Little Security in an Insecure World" [***PBDJ***, Vol. 9, issue 12]) established a level playing field by introducing you to terminology that as PB developers we typically don't deal with on a daily basis. One of the key points in Part 1 was that there is more to security than just hiding something behind a user ID and a password. PowerBuilder does not expose a robust cryptographic library, meaning critical security functionality like encryption, secure hashing, and even rudimentary Base64 encoding capabilities isn't easily available.

The world of Java exposes many cryptographic capabilities from inside the Java Cryptography Extensions library (JCE). The JCE can be configured to work with JDK 1.2.2 and beyond; see http://java.sun.com for detailed specifications. What that means to us PowerBuilder developers is that with a little configuration, it is possible to employ the techniques covered herein using any combination of PowerBuilder 7.0 and beyond with EAServer 3.6.1 (running no less than JDK 1.2.2) and beyond. As I sit here in my favorite coffee establishment drinking a Caramel Macchiato, my laptop is configured solely with PowerBuilder 8.0 and EAServer 4.1.3 running JDK 1.4.

I feel compelled to remind everyone that the code examples here are just that, examples, and I leave it up to each of you to determine the applicability of using this code in a production environment! I should also mention that at present, it isn't possible to easily access Java libraries from a traditional two-tier client/server program. PowerBuilder 9.0 exposes the new PowerBuilder Native Interface, and I've already begun to see some of my peers working on examples that demonstrate how to access Java code directly from PowerBuilder without requiring a bridge like Sybase EAServer in the middle. Until the release of PowerBuilder 9.0, we'll have to stick with the EAServer-in-the-middle approach

Clearly, I don't have room here to provide an in-depth explanation of the Java language, the build and deployment process, use of the JCE APIs, and accessing the final EAServer components from within PowerBuilder. Instead, we'll focus on what is required to properly configure EAServer to access the JCE and the API exposed by the EAServer component I've written (available at www.sys-con.com/pbdj/sourcec.cfm), and call the cryptographic methods from within a PowerBuilder application.

A View into the PBSecurity/Crypto EAServer Component

Let's start by downloading and unzipping the pbsecurity.zip file, available from the *PBDJ* Web site. For those with limited exposure to Java, let me say up front that the .zip file contains a precompiled, properly formatted EJB .jar file at pbdj/lib/pbsecurity.jar that's ready for deployment into EAServer 4.1.x. The Crypto EJB component is comprised of three Java classes and two Java interfaces. To be more concise, it's actually a single stateless session EJB with two utility classes. I intentionally placed all the cryptographic operations in the standalone utility class com.getpersonified.pbdj.CryptOps so that you would have the opportunity to reuse this cryptographic code outside of an EJB container. The Base64 code is included for accessibility, but was originally written by Robert Harder and downloaded from http://iharder.sourceforge.net/base64. A simple review of the EJB's implementation class com.getpersonified.pbdj.CryptoBean reveals that the exposed cryptographic methods are merely wrapper calls out to either the Base64 or CryptOps utility class.

In essence, the EJB provides the necessary methods to perform Base64 encoding/decoding, RSA key pair generations for use in asymmetric cryptography, and production of an SHA-1 message digest (secure hash). I'll leave it up to you to extend the methods, adding support for additional symmetric algorithms and such.

Now let's get EAServer set up to perform some cryptography!

Printable Characters

It's probably worth reminding everyone that in the ASCII character set, not every character is printable. As a result, especially when dealing with cryptographic operations, all the results displayed and returned from the component I've developed are automatically Base64 encoded for readability and easy storage, for example, in a database column using a varchar or similar datatype.

Choose Your Cryptographic Provider

Sun's JDK ships with a small set of cryptographic implementations and, in fact, doesn't provide

any asymmetric algorithms, like the industry-dominant RSA algorithms. In fact, many Java cryptology experts recommend avoiding Sun's JCE provider altogether because once the Sun provider is loaded, it prevents the use of other providers (see *Professional Java Security* by Jess Garms and Daniel Somerfield for more details).

Since the PKI operations we want to demonstrate require public and private key generation, we need to look beyond the Sun JCE provider for an alternate that supports asymmetric algorithms. One of the best JCE providers that I've worked with is from The Legion of the Bouncy Castle (http://bouncycastle.org), and that's what we'll set up.

You'll need to download two .jar files from the legion at www.bouncycastle.org/latest_releases.html. The first .jar file is the JDK 1.4 Provider under Signed Jar Files header, named something like bcprov-jdk14-116.jar. Once you have this .jar file, you'll need to install it as an extension into the JVM that EAServer uses. By default, EAServer contains its own JDK directories in the $SYBASE_SHARED/Sun/jdk directory. The JVM contains an "extensions" directory in which every .jar file dropped into the directory is automatically made available to the JVM without having to make any BOOTCLASSPATH or CLASSPATH modifications. Be especially careful and never arbitrarily drop a .jar file into this location, as it's assumed that only trusted code would ever be placed here!

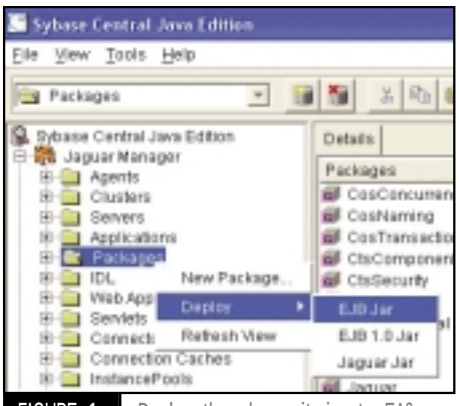


FIGURE 1 Deploy the pbsecurity.jar to EAServer

The next .jar file you'll need is on the same page, named something like jce-jdk13-116.jar under the Clean Room JCE and Provider header.

• ***Step 1:*** Copy the bcprov-jdk14-116.jar file into the $SYBASE_SHARED/Sun/jdk/jdk1.4/jre/lib/ext directory on your machine. Copy the jce-jdk13-116.jar into your $JAGUAR root directory.

If you've used EAServer for some time, you're familiar with some of the classloader idiosyncrasies it displays. This is one of them. Because of article deadlines, I didn't have time to fully explore why this configuration works, but it does. Despite placing the Bouncy Castle provider into the extensions folder of the JDK, and the fact that JDK 1.4 is supposed to ship with the JCE, EAServer 4.1.3 under JDK 1.4 requires the addition of the jce-jdk13-116.jar file to its CLASSPATH and BOOTCLASSPATH configurations in its $JAGUAR/bin/serverstart.bat. Not adding this .jar file in the aforementioned paths results in a NoClassDefFoundException like that listed below when encryption and decryption is attempted from our EJB:

```
java.lang.NoClassDefFoundError: javax/
 crypto/Cipher
 at com.getpersonified.pbdj.CryptOps.encrypt
  (CryptOps.java:58)
 at com.getpersonified.pbdj.CryptoBean.
  encrypt(CryptoBean.java:52)
 at com.getpersonified.pbdj._sk_PBSecurity_
  Crypto.invoke(_sk_PBSecurity_Cry…
```

In EAServer 4.1.3, the CLASSPATH and BOOTCLASSPATH are configured around lines 417 and 468, respectively.

• ***Step 1.5:*** Add the following lines after the respective "." configuration lines:

```
set EAS_CLASSPATH_P1=.
set EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;
 %JAGUAR%\jce-jdk13-115.jar
```

and

```
set EAS_BOOTCLASSPATH_P1=.
set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASS
 PATH_P1%;%JAGUAR%\jce-jdk13-115.jar
```

I fired off several e-mails to Sybase engineers, but with the holiday break I hadn't received a response prior to my submission deadline as to why this extra .jar might be necessary.

Exposing Cryptographic Providers to the JVM

With our JCE provider class files now in place and EAServer's idiosyncrasies addressed, we need to inform the JVM that it should include the Bouncy Castle provider class in its search path for cryptography algorithm implementations. The architecture of the JDK inspects a file named java.security that lives inside the $JDK/jre/lib/security directory to locate the names of known JCE providers.

• ***Step 2:*** Open up the $SYBASE_SHARED/Sun/jdk/jdk1.4/jre/lib/security/java.security file using your favorite text editor. After a large comment block at the top, there's a section header that reads:

```
#
# List of providers and their preference
orders (see above):
#
```

We want to modify this section to include a reference to our Bouncy Castle provider class. To emphasize the subtlety of the title, the order these classes are listed in is very important. Modify your list to read:

```
#
# List of providers and their preference
orders (see above):
#
security.provider.1=sun.security.provider.
 Sun
security.provider.2=com.sun.rsajca.Provider
security.provider.3=org.bouncycastle.jce.
  provider.BouncyCastleProvider
security.provider.4=com.sun.net.ssl.inter
  nal.ssl.Provider
#security.provider.4=com.sun.crypto.provider
  .SunJCE
security.provider.5=sun.security.jgss.
  SunProvider
```

The JCE uses a factory design pattern, and what we've done is give a list of providers that the JCE should search (in order) when a specific cryptographic algorithm is requested. A factory design pattern takes a string as input that represents the name of a class that implements a well-defined interface or extends a well-known super class, returning an instance of the named class appropriately cast. While this is an oversimplification of what actually goes on, suffice it to say that it allows code to be written that doesn't form a dependency on the class being instantiated. As long as the named class implements the well-defined interface, the factory can dynamically find, load, and cast the class to a well-known interface. By adding the Bouncy Castle provider to this list, we can now request and instantiate the asymmetric algorithms that we need to perform public key cryptography. Consider this line of Java code from the CryptOps class:

```
KeyPairGenerator keyPairGenerator =
KeyPairGenerator.getInstance("RSA");
```

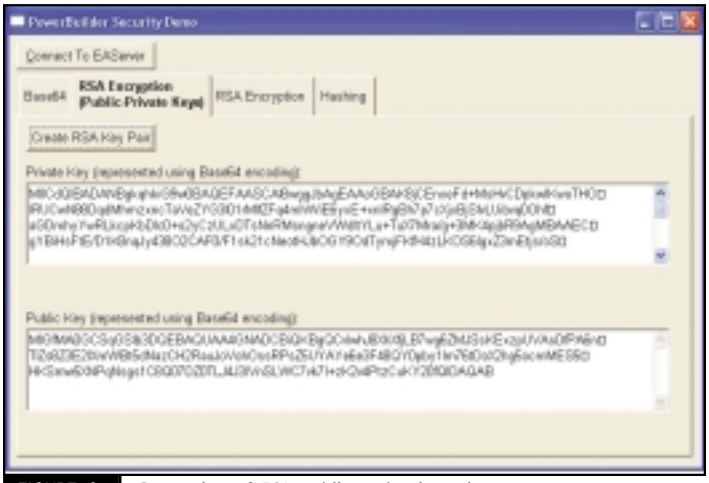


FIGURE 2 Generation of RSA public and private keys

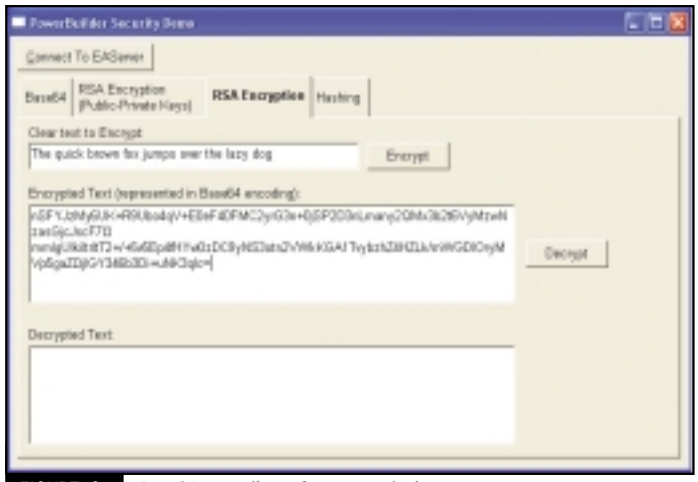


FIGURE 3 Base64 encoding of encrypted clear text

The KeyPairGenerator class is the factory we've been talking about, and getInstance() is the static method on that class that does the actual work. At runtime, it looks up the concrete class that provides RSA asymmetric functionality and instantiates it. This method interrogates each of the listed providers in java.security, in sequence, asking them if they can serve up a KeyPairGenerator named RSA. Once the JCE finds our Bouncy Castle provider in the list, an RSA KeyPairGenerator is instantiated. As a result of this design, new cryptographic algorithms can be added at anytime without recompiling or redistributing the core JVM classes.

Building the PBSecurity/Crypto Component

For those who like to tinker and are familiar with Apache Ant, I've included the Ant script I used to build and deploy the EJB. Running the default target displays the following help:

```
C:\starbucks\pbdj\pbdj>ant
Buildfile: build.xml

help:
     [echo]
Welcome to the JCE security for
PowerBuilder ant script. This script is
used to build the entire JCE EJB example.
The following tasks are available

build    - builds the JCE EJBs that
           PowerBuilder will call
deploy   - deploys the JCE EJBs to EAServer
javadoc - builds JavaDoc documentation for
           the EJB


BUILD SUCCESSFUL

Total time: 2 seconds
```

In essence, the three targets will build, deploy, and document the component. In fact, the easiest way to deploy the pbsecurity.jar file is through the deploy target. Be sure to verify your configuration before running the deploy target by inspecting lines 21 through 24 that set up where EAServer is running:

```
<property name="jaguar.host" value="local
  host" />
<property name="jaguar.port" value="9000"
  />
<property name="jaguar.user" value="jagad
  min" />
<property name="jaguar.password"
  value="dev" />
```

Deploying the PBSecurity/Crypto Component

For those who are anxious to just deploy the code and not tinker with Apache Ant, the .zip file contains an EJB .jar file ready for immediate deployment. There is no additional configuration necessary beyond the installation of the Bouncy Castle provider covered earlier.

- ***Step 3:*** Connect to EAServer using Jaguar Manager, right-click on the Packages folder, and choose deploy EJB, as shown in Figure 1.

In the resulting dialog, enter the full path (or browse) to the $PBDJ/lib/pbsecurity.jar file included with the pbsecurity.zip file you downloaded.

Note: Be sure to install the package into EAServer as well by right-clicking on the Installed Packages folder under the Jaguar server and choosing the Install Package… menu item.

PBSecurity PowerBuilder Client Application Overview

Time to fire up our trusty PowerBuilder IDE. Included in the pbsecurity.zip file is a workspace named securityclient.pbw that should automatically include the securityclient.pbl as a target when opened. The client application is a single window (no flames about the GUI design – it's a demo!) that contains a hidden tab control until you successfully establish a connection with EAServer.

Note: Be sure to review the code in the Open! event of w_client to ensure it points to your EAServer installation.

Once a connection to EAServer is established, the tab control is made visible. There are four tabs that demonstrate the different aspects of security we've been talking about, including Base64 encoding, public/private key generation, asymmetric encryption, and finally secure hashing.

Digging into the PowerBuilder Code

You'll notice that the library contains two proxy objects, crypto and cryptohome, as well as two supporting structures, cryptolist and ctscomponents_stringseq. These four objects represent the entire set of objects necessary to provide cryptographic capabilities from inside PowerBuilder.

Each of the operations on the client window follow the same basic approach – look up the home proxy object (cryptohome), get an instance of the proxy for the bean (crypto), and invoke the desired method:

```
CryptoHome lnv_home
Crypto lnv_crypto

icn_jag.lookup(lnv_home, "PBSecurity/
 Crypto","com/getpersonified/pbdj/
 CryptoHome")
lnv_crypto = lnv_home.create()

String base64Text
base64Text = lnv_crypto.stringtobase64
 (sle_1.text)
```

Of particular interest is that in order to perform the public key encryption, you must first obtain a set of keys. Feel free to hit the Create button a couple of times, comparing the time it takes to generate the first key pair to the time it takes to generate the second through *n*th key pairs. That first time, the cryptographic engine is priming itself with random data from your system, thus the added delay can't be avoided. Remember, the methods in the EJB automatically Base64 encode the keys so they can easily be displayed. In Figure 2, we see that we've obtained public and private RSA keys.

To reiterate an important point, in PKI the public key can be distributed freely, while the private key should remain in a secure location and never be disclosed. Once you've generated

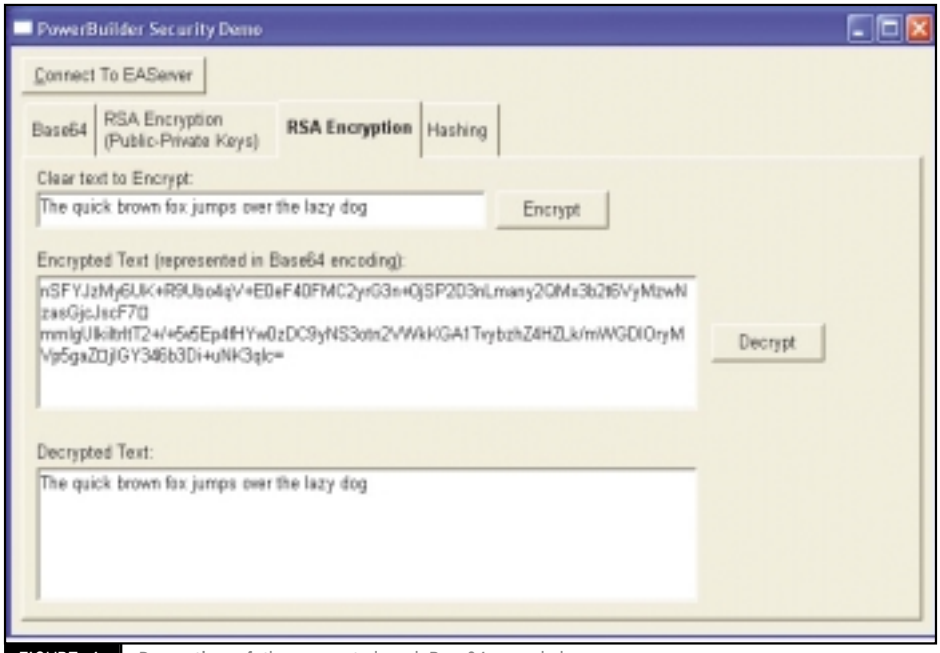


FIGURE 4 Decryption of the encrypted and Base64 encoded message

the key pair, the RSA Encryption tab becomes enabled. Clicking on that tab reveals a single line edit where you can enter in your clear text message that you want to encrypt. In Figure 3, we can see that our clear text message was successfully encrypted and then Base64 encoded for display. Encryption was performed using the RSA public key, as emphasized by the message boxes that display during the process. You might want to try and copy the Base64 encoded encrypted message and paste it into decode single line edit on the Base64 page. Be prepared to see a bunch of funny characters and hear a couple of system beeps as the nonprintable and control characters are, well, printed! This should serve as a sanity check that we did more than just Base64 encode the clear text message.

To decrypt the message, we submit the Base64 encoded text along with the RSA private key. Mathematically speaking, only the private key can successfully decrypt text encrypted with the public key (and vice versa). The decrypted text is then displayed as shown in Figure 4.

The other two tabs, Base64 and Hashing, are fairly self-explanatory so I won't cover how they're used here.

Conclusion

Having access to cryptographic operations is critical in this day and age. I've demonstrated how the combination of PowerBuilder and EAServer can be utilized to expose advanced cryptographic operations to PowerBuilder applications and PowerBuilder components. Using the sample code, I encourage all of you to set off and play with the various cryptographic topics we covered in Part 1. Until next time, I say to all of you: R28gQnVja2V5ZXMt LSBCZWF0IE1pY2hpZ2FuIQ==. ▼

Author Bio

Jason Weiss is the chief software architect at Personified Technologies, LLC, where he spends his days architecting digital rights management (DRM) solutions for clients.

jason@getpersonified.com.

SHOP ONLINE AT JDJSTORE.COM FOR BEST PRICES OR CALL YOUR ORDER IN AT 1-888-303-JAVA

JDJStore.com

GUARANTEED LOWEST PRICES!

Attention Java Vendors:
To include your product in JDJStore.com, please contact Rachel@sys-con.com

POINTBASE
$353.00 PointBase Embedded 4.4 - Windows Installer
PointBase® Embedded is a 100% Pure Java RDBMS ideal for embedding in your Java applications, within the J2EE and J2SE platforms. PointBase Embedded utilizes a multi-threaded architecture allowing multiple connections from within the same JVM. PointBase Embedded features also include small footprint architecture, extensibility, referential integrity, near-zero administration, and a low cost of ownership.

SYBASE
$2,558.99 PowerBuilder Enterprise v.8
Introducing PowerBuilder 8! The award-winning, industry-leading application development environment just got better. Release 8 of PowerBuilder adds exciting new features and capabilities that will make your development of Web, client/server, and distributed applications easier, faster, and more cost-effective. New features focusing on improved productivity, tighter integration with EAServer, and Web application development make this release indispensable to your development efforts.

/N SOFTWARE
$295.00 IP*Works! SSL Scripting Edition (1 cpu)
IP*Works! SSL adds SSL and Digital Certificate capabilities to the IP*Works! Internet Toolkit providing Secure Web Browsing, Secure Client, Secure Server, Secure Mail, Digital Certificate Management Capabilities, and a lot more. The current release consists of 15 components for secure Internet connectivity.

ACCELTREE
$469.99 FULCRUM Professional Edition 1.1 - A Java Code Assembler
"FULCRUM" is a Java development tool that uses a proprietary concept of code templates that can be used as "building blocks" to construct efficient Java objects and applications. Unlike code generators that impose their coding techniques and run-time environments on the users, FULCRUM allow users to build their own coding standards in the templates and also lets users switch to manual coding at any point in time.

SIMPLICITY
$3,999.00 Simplicity Enterprise Lite
Simplicity Enterprise Lite provides the ability to build server-side, enterprise class Web applications. Using a palette of graphical modules, the developer links together their desired functionality and interactively tests their application in one seamless development environment. Application targets include Web sites, WAP-enabled mobile devices, *n*-tier application services, and enterprise Servlets.

BORLAND
$389.99 JBuilder 7 SE NEW-USER 98/W2K/NT/LINUX/SOL
JBuilder is the leading cross-platform environment for learning Java programming and personal application development. JBuilder 6 Personal includes an integrated editor, debugger, compiler, visual designers, wizards, and tutorials.

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

WWW.JDJSTORE.COM

Dynamic Report Module Basics

Take your PB apps to the next level

WRITTEN BY
DAVID HART

Now that the holidays are over, it's back to work, back to school…and back to the basics. In my December column, "PowerBuilder and ODBC 101" (*PBDJ*, Vol. 9, issue 12), we discussed hooking up to a database via ODBC.

This month, let's look at the most important item to hook up to a database, a DataWindow. Our apps are all about viewing data, and the DataWindow is the nucleus of all "data viewing" in PowerBuilder apps. (Source code for this article can be downloaded from www.sys-con.com/pbdj/sourcec.cfm.)

Since most DataWindows function as "reports," we'll use that as our model. Let's pretend our users want a window to view "reports." We can put this together using one window, one DataWindow control, one listbox, one object-level function, a few DataWindow objects, about a dozen lines of code, and some elbow grease. Only a dozen lines of code? We only need *one* DataWindow control? Yes, that's all we need if we use the dataobject property of the DW control dynamically at runtime.

Most folks starting out with PB get into a habit of using one DW control with one DW object defined at design time. Good for starting out, but not very flexible. With a little creativity, we can have so much more, for so much less. Fire up the new PC you got in December from your rich relatives and follow along with me:

- Create a new workspace and target or just use ones you already have.
- Create a main! window called "w_reports". Drop a DataWindow control and a listbox onto it. Leave the DW control named dw_1 and change the name of the listbox to lb_reports. Save the window.
- Create three DataWindow objects using the sample database (EAS Demo DB V4).
  1. ***dw_employee:*** A simple report showing all employees in alphabetical order
  2. ***dw_customer:*** A simple report showing all customers in alphabetical order
  3. ***dw_phonebook:*** A join between employees and department, grouped by department

At this point, we have all the basic pieces to put together a report module. Let's review the steps required to get a DW "talking" to the database.

1. Initialize a transaction object (your own) or use SQLCA.
2. Populate the transaction object properties needed for your database.
3. Try connecting to the database.
4. Check the transaction object's SqlCode property to see if the connection was successful.
5. Set the dataobject property of the DataWindow control at design time or programmatically (dw_1.dataobject = '<dw object name>').
6. Connect the DW control to your transaction object (dw_1.SetTransObject (<tranx object name>). Be sure to check the return code for a successful connection.

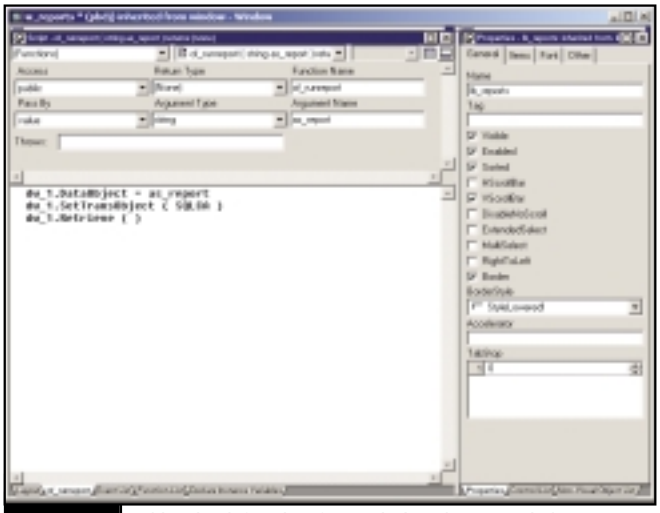


FIGURE 1 Object-level function for retrieving the DataWindow result set

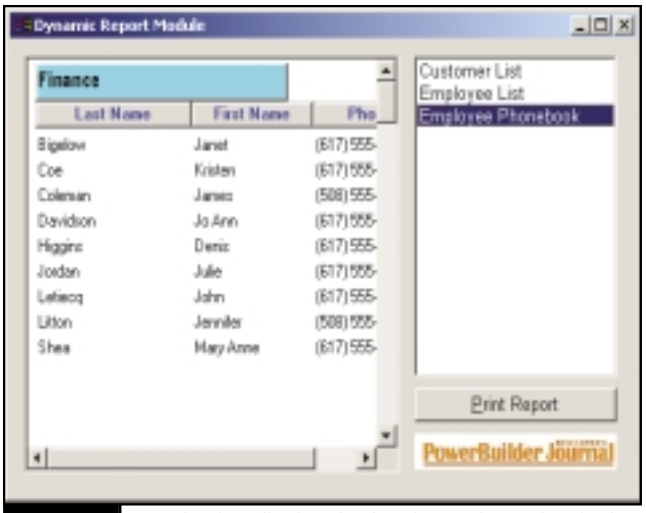


FIGURE 2 Completed application showing the Employee Phonebook

> "Changing properties 'on the fly' (at runtime) is a key concept to master to take your PB apps to the next level"

7. Perform a Retrieve() or Update() on the DW control.

I'll assume you already have SQLCA successfully connected to the database (if you don't, review my December article). We must use the <dw_control>.SetTransObject (<tranx object>) function. However, we must remember the following rule: if we change the dataobject property, the connection is broken and we must reset the DW control to the transaction object. We're going to do just that: change the dataobject property "on the fly" in our app, giving us a very flexible report tool. Let's start coding:

- Open w_reports in the window painter.
- Declare an instance variable of type DataWindow:

```
datawindow        idw_report
```

  - Make sure you save the window before leaving the instance variable area.
- Create an object-level function called "of_runreport". Make it with one input parameter of type DataWindow. Code it as shown in Figure 1.
- Select the listbox lb_reports. On the Properties panel, select the Items tab. We'll make our report "pick list". Create the items as follows:
  - Customer List
  - Employee List
  - Employee Phonebook
- Code the SelectionChanged event of lb_reports to determine which report was selected by the user and retrieve it as follows:

```
CHOOSE CASE index
 CASE 1
  // Customer List
   of_runreport( 'dw_customer' )
 CASE 2
  // Employee List
   of_runreport( 'dw_employee' )
 CASE 3
  // Employee Phonebook
   of_runreport( 'dw_phonebook' )
END CHOOSE
```

Run the application. The DW control will be empty initially. Just click on one of the reports. Each time you select a report, it will change the DW control dataobject property, reset the transaction object connection, and retrieve the data. Figure 2 shows the completed window. By using one DW control, the print button logic is simple: dw_1.Print ( ). Now you have a base report module. You may want to add a few "bells and whistles," such as radio buttons, to re-sort the data from ascending to descending, a button for filtering, and a button that turns on the "prompt for criteria" functionality.

Changing properties "on the fly" (at runtime) is a key concept to master to take your PB apps to the next level. Happy coding! ▼

dhart@niteshift.com

Author Bio

David Hart is a certified PB and ASE instructor/developer. He resides in Rancho Cucamonga, California, and is putting the finishing touches on his first book, PowerBuilder 101: A Practical Way to Learn PB.

PBDJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
.NET Developer's Journal	www.sys-con.com/dotnet/	888-303-5282	19
Amyuni Technologies, Inc.	www.amyuni.com	866-926-9864	2
E.Crane Computing	www.ecrane.com		35
Java Developer's Journal	www.javadevelopersjournal.com	888-303-5282	23
JDJ Store	www.jdjstore.com	888-303-JAVA	27
PB9 Books	www.pb9books.com		15
PowerBuilder Developer's Journal	www.sys-con.com	888-303-5282	21
Sybase, Inc	www.sybase.com/pbextension		5
Sybase, Inc	www.sybase.com	800-8-SYBASE	36
SYS-CON Media	http://developer.sys-con.com	888-303-5282	19
SYS-CON Subscription Offer	www.sys-con.com/suboffer.cfm	888-303-5282	32, 33
Web Logic Developer's Journal	www.weblogicdevelopersjournal.com	888-303-5282	29
Web Services Edge 2003	www.sys-con.com	201-802-3069	9
Web Services Journal	www.wsj2.com	888-303-5282	11

Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

THE WORLD'S LEADING INDEPENDENT WEBLOGIC DEVELOPER RESOURCE

SPECIAL OFFER! SAVE $31* OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Helping you enable intercompany collaboration on a global scale

- Product Reviews
- Case Studies
- Tips, Tricks and more!

Now in More than 5,000 Bookstores Worldwide – Subscribe *NOW!*

Go Online & Subscribe Today!

*Only $149 for 1 year (12 issues) – regular price $180.

WebLogicDevelopersJournal.com

SYS-CON Media, the world's leading publisher of *i*-technology magazines for developers, software architects, and e-commerce professionals, brings you the most comprehensive coverage of WebLogic.

SYS-CON MEDIA

Using Oracle Stored Procedures in Your DataWindows

Powerful and flexible

WRITTEN BY
BRUCE ARMSTRONG

Oracle was slow to introduce support for returning result sets from stored procedures. When they finally did add such support, they did so in a manner that was somewhat different than the approach used by other vendors.

This has a couple of consequences for PowerBuilder developers. The first is that the PBDBMS approach was developed within PowerBuilder as an interim workaround. The second is that using stored procedures to return result sets is not as widespread a practice with Oracle databases as it is with other database products.

This article (an excerpt from an upcoming book on PowerBuilder 9) is an attempt to address this situation. The interim PBDBMS solution is going away. In PowerBuilder 8, support for the PBDBMS was restricted to the O73 driver and with PowerBuilder 9, support for it will be dropped completely. If you're still using that old technique and need to migrate, or you would just like to learn the new one, this article should help.

Using a Stored Procedure for the SQL Source for a DataWindow

While a SQL SELECT statement is the typical way data is retrieved from the database into a DataWindow, there is another approach. Using a stored procedure to do the retrieve is popular with a number of other databases because they offer some advantages over a standard SELECT statement. First, stored procedures in the database are precompiled, which means they'll significantly outperform uncompiled SQL statements. In addition, stored procedures provide for data-hiding, i.e., the end user doesn't need to know or have the ability to access the underlying tables in order to retrieve the data.

Note that neither of these advantages are significant when using an Oracle database. Provided the application is being run with binding enabled, the database is storing the bound and compiled statements in its SHARED_SQL area, which means that SQL SELECT statements will achieve similar performance as stored procedures. Further, Oracle's role-based security model can and should be used to restrict the user's ability to access data only in the context of the application session. That is, the role they need to access the data can be enabled only for the application session, and only for the duration of the application session (nondefault roles expire when the session ends).

Regardless, using stored procedures for the data source is an option, and may particularly be preferred for consistency when the application has to support a number of different database vendors, and stored procedures are used as the source for other vendors.

Ref Cursors

Oracle introduced the concept of ref cursors (a cursor that can be passed back to a client) with version 7.2. Prior to that, Sybase provided a workaround in PowerBuilder known as PBDBMS (named after the package created by PowerBuilder in the database used to return the data) to enable you to return a result set from an Oracle stored procedure. Support for that feature is being dropped and its use should be discontinued. Therefore our discussion will focus entirely on the ref cursor approach.

To return a ref cursor, it must first be declared as a data type, generally in a package specification. A ref cursor can be constrained, which means that the result set description is defined when the ref cursor is declared, or it may be unconstrained, which means that the result set description is determined upon use. PowerBuilder supports the use of either, but it does bind the DataWindow to the particular result set description it obtained when the DataWindow was created through the execution of the stored procedure. Therefore, while both are supported, the use of an unconstrained ref cursor could lead to runtime errors. This is particularly true because the developer could inadvertently modify the result set description after the initial creation of the DataWindow, and the unconstrained ref cursor would allow the modified statement to compile. In the case of a constrained ref cursor, a modification to the result set without modification of the ref cursor data type declaration would result in an error during the stored procedure compile, alerting the developer to the issue. On the other hand, there are some techniques – particularly using object types to store and return the data – that necessitate the use of an unconstrained ref cursor.

The ref cursor can be populated by a number of methods, and a straight select statement is the most common. However, there are also methods available to handle intermediate data. With some other databases, this is accomplished through the use of temporary tables. Much the same can be done with Oracle, except that instead of creating a physical temporary table, we create something similar in memory. We'll use the code in Listing 1 for our examples.

Note that PL/SQL record types can also be declared to be equivalent to a table definition through the %ROWTYPE attribute. So we could have used this instead:

```
TYPE r_employee IS emp%rowtype ;
```

Or more simply:

```
TYPE constrainedcursor IS REF CURSOR
RETURN emp%rowtype ;
```

The actual element-by-element record type declaration is used in the examples

This article is based on *PowerBuilder 9 Client/Server Development* by various authors (ISBN 0672325004), which will be published by Sams Publishing in March 2003. Also look for *PowerBuilder 9 Internet and Distributed Application Development*, coming in March 2003.

in Listing 2 because it offers greater flexibility with regard to mixing elements from more than one table (or elements that are independent of any table) though it does require a bit more work.

Before we show the unconstrained example, we'll create a couple of object types first.

```
CREATE OR REPLACE TYPE EMPINFO IS
OBJECT (
 empno    number(4,0),
 ename    varchar2(10),
 job varchar2(9)
 )  ;

CREATE OR REPLACE TYPE EMPINFOLIST AS
TABLE OF EMPINFO ;
```

We're going to use those object types as a method of storing intermediate data. Note that we used actual data types rather than column type references to create the object type (see Listing 3). (Listings 3–5 can be downloaded from www.sys-con.com/pbdj/sourcec.cfm.)

The "TABLE" operator was introduced with Oracle8*i* as a replacement for the "THE" operator in Oracle 8.0. If you're working with an Oracle 8 database, the stored procedure would be modified as follows:

```
-- Finally, let's pass back the data
OPEN c_empinfo FOR
 SELECT *
 FROM THE ( SELECT ( CAST ( v_empin-
folist AS empinfolist ) ) FROM DUAL )
;
```

Either approach requires object support to be enabled on the server. In Oracle 8 you must have the enterprise edition in order to use objects. 8*i* and later support objects in the standard edition. To verify whether you have support for objects, use the following query, which will return TRUE or FALSE to indicate if the option is installed:

```
select value from v$option where
parameter = 'Objects'
```

If object support is not enabled on the server or if you're working with a 7.2 or 7.3 database (which does not have the objects option), we can use something more like a physical temp table approach. We'll need to create a table that will define our result set, just like the object type would:

```
CREATE TABLE TEMP_EMPINFO (
 empno            number(4,0),
 ename            varchar2(10),
 job              varchar2(9) ) ;
```

We'll modify our stored procedure as shown in Listing 4.

Although our temp table in this case is permanent, we don't need to track the session ID in it because we aren't performing a COMMIT without our procedure. The only data we'll see (and delete) in the table is the data we've put there within the session.

Beginning with Oracle8*i*, support for temporary tables was introduced with the "GLOBAL TEMPORARY" keyword pair for the CREATE TABLE DDL statement. It basically does the same thing we've accomplished here.

Using Stored Procedures for DataWindow Updates

If you're going to be using stored procedures to query the database, either you won't allow updates from the DataWindow (it's a report) or you'll allow updates and there's a good chance you'll be using stored procedures for the updates as well. If you're using the stored procedure to provide for information hiding (preventing the user from selecting directly off the base tables), it doesn't make sense to allow them to have direct update capabilities on those same tables.

We'll add a couple more procedures to our package (see Listing 5), and use them for the update specs (see Figure 1).

Conclusion

Hopefully, this article will have removed some of the mystery concerning how to use Oracle stored procedures to return result sets. While the server-side technique differs considerably from that used by other vendors and the interim PBDMS approach, I believe you'll find it flexible and powerful once you get comfortable using it. ▼

bruce.armstrong@teamsybase.com

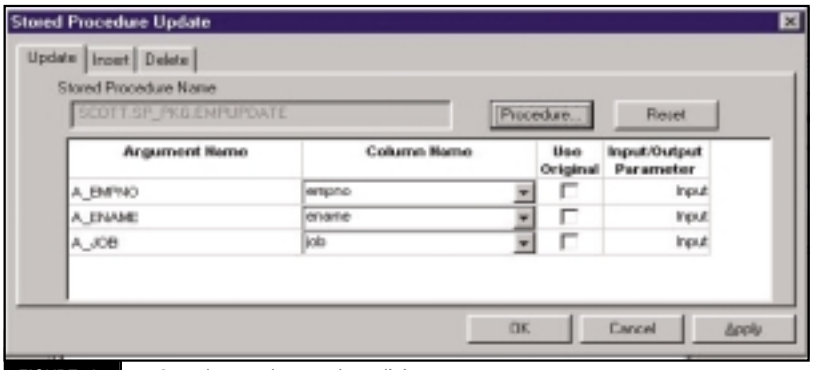


FIGURE 1 Stored procedure update dialog

Author Bio

Bruce Armstrong is a senior PowerBuilder developer with the Los Angeles office of Kforce.com. A member of TeamSybase, he has been using PowerBuilder since version 1.0.B. He was a contributing author to PowerBuilder 4.0 Secrets of the Masters. *He is also a member of the editorial board for* PowerBuilder Developer's Journal *and the author of several* PBDJ *articles, including a monthly news column.*

Listing 1

```
CREATE OR REPLACE PACKAGE SP_PKG AS
-- A PL/SQL record type
TYPE r_employee IS RECORD (
 empno emp.empno%type,
 ename emp.ename%type,
 job emp.job%type ) ;

-- A constrained cursor (we use the record  type to define
our result set)
TYPE constrainedcursor IS REF CURSOR RETURN r_employee ;
-- An unconstrained cursor (no definition of result set in
declaration)
TYPE unconstrainedcursor IS REF CURSOR ;

-- A stored procedure returning our constrained type
PROCEDURE callconstrained (
 a_dept  IN  emp.deptno%type,
 c_emp OUT constrainedcursor ) ;

-- A stored procedure returning our unconstrained type
PROCEDURE callunconstrained (
 a_dept  IN emp.deptno%type,
 c_empinfo OUT unconstrainedcursor ) ;

END SP_PKG ;
```

Listing 2

```
CREATE OR REPLACE PACKAGE BODY SP_PKG AS
PROCEDURE callconstrained (
 a_dept  IN  emp.deptno%type,
 c_emp OUT constrainedcursor ) IS
BEGIN
 -- About the simplest thing we could do
 OPEN c_emp FOR
  SELECT empno,
   ename,
   job
   FROM emp
  WHERE deptno = a_dept ;
END callconstrained ;
--
--<some other code>
--
END SP_PKG ;
```

Download the Code!

The code listing for this article can also be located at www.sys-con.com/pbdj/

A LIMITED TIME SAVINGS

SUBSCRIBE TODAY TO MULTI

Go To www.sys-con.com/suboffer.cfm

and receive your **FREE** CD Gift Package VIA Priority Mail

Each CD is an invaluable developer resource packed with important articles and useful source code!

Pick the CDs to go with your Multi-Pack order

- Pick one CD with your 3-Pack order
- Pick two CDs with your 6-Pack order
- Pick three CDs with your 9-Pack order

- ☐ Web Services Resource CD
- ☐ Java Resource CD
- ☐ WebLogic Resource CD
- ☐ ColdFusion Resource CD
- ☐ XML Resource CD
- ☐ WebSphere Resource CD
- ☐ CF Advisor Complete Works CD

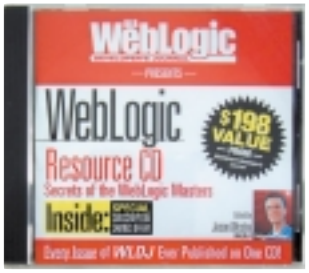
Your order will be processed the same day!

More than 1,400 Web services and Java articles on one CD! Edited by well-known editors-in-chief Sean Rhody and Alan Williamson, these articles are organized into more than 50 chapters on UDDI, distributed computing, e-business, applets, SOAP, and many other topics. Plus, editorials, interviews, tips and techniques!

LIST PRICE $198

The most complete library of exclusive *JDJ* articles compiled on one CD! Assembled by *JDJ* Editor-in-Chief Alan Williamson, more than 1,400 exclusive articles are organized into over 50 chapters, including fundamentals, applets, advanced Java topics, Swing, security, wireless Java,... and much more!

LIST PRICE $198



The most complete library of exclusive *WLDJ* articles ever assembled! More than 200 articles provide invaluable information on "everything WebLogic", including WebLogic Server, WebLogic Portal, WebLogic Platform, WebLogic Workshop, Web services, security, migration, integration, performance, training...

LIST PRICE $198

The most complete library of exclusive *CFDJ* articles on one CD! This CD, edited by *CFDJ* editor-in-chief Robert Diamond, is organized into more than 30 chapters with more than 400 exclusive articles on CF applications, custom tags, database, e-commerce, Spectra, enterprise CF, error handling, WDDX... and more!

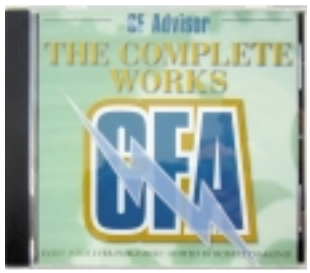
LIST PRICE $198

The largest and most complete library of exclusive *XML-Journal* articles compiled on one CD! Edited by well-known editors-in-chief Ajit Sagar and John Evdemon, these articles are organized into more than 30 chapters containing more than 1,150 articles on Java & XML, XML & XSLT, <e-BizML>, data transition... and more!

LIST PRICE $198

The most up-to-date collection of exclusive *WSDJ* articles! More than 200 articles offer insights into all areas of WebSphere, including Portal, components, integration, tools, hardware, management, sites, wireless programming, best practices, migration...

LIST PRICE $198



The complete works of *CF Advisor* are now on one CD! Check out over 200 exclusive articles on topics such as e-commerce, custom tags, Fusebox, databases, object-oriented CF, WDS, upgrading CF, wireless, Verity, and more. Plus, find interviews, editorials, and source code!

LIST PRICE $198

Subscribe Online Today

PLE MAGAZINES ONLINE

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

Pick a 3-Pack, a 6-Pack or a 9-Pack ☑

TO ORDER: Choose the Multi-Pack you want to order by checking next to it below. Check the number of years you want to order. Indicate your location by checking either US, Canada/Mexico or International. Then choose which magazines you want to include with your Multi-Pack order.

☐ 3-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.
☐ 6-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.
☐ 9-Pack	☐ 1YR	☐ 2YR	☐ US	☐ Can/Mex	☐ Intl.

☐ **Java Developer's Journal**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. -One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **XML-Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **WebLogic Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **Wireless Business & Technology**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **WebSphere Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

3-Pack

Pick any 3 of our magazines and save up to $275.00

Pay only $175 for a 1 year subscription plus a **FREE CD**

- *2 Year – $299.00*
- *Can/Mex – $245.00*
- *All Other Cnty. – $315.00*

6-Pack

Pick any 6 of our magazines and save up to $350.00

Pay only $395 for a 1 year subscription plus **2 FREE CDs**

- *2 Year – $669.00*
- *Can/Mex – $555.00*
- *All Other Cnty. – $710.00*

9-Pack

Pick all 9 of our magazines and save up to $400.00

Pay only $400 for a 1 year subscription plus **3 FREE CDs**

- *2 Year – $839.00*
- *Can/Mex – $695.00*
- *All Other Cnty. – $890.00*

SYS-CON MEDIA

PowerBuilder News

All things of interest to the PB community

BY BRUCE ARMSTRONG

CORPORATE

AvantGo

12/20/02 – Sybase and AvantGo, a leading provider of mobile enterprise software, announced that they have signed a definitive agreement for Sybase to acquire AvantGo in a cash merger valued at approximately $38 million.

Sybase intends to operate AvantGo under its *i*Anywhere Solutions subsidiary. The acquisition is expected to be completed during the first quarter of 2003. www.avantgo.com

Intraware

12/19/02 – Intraware announced that Sybase selected the Intraware SubscribeNet solution to provide electronic delivery of product releases to the Sybase global customer base. Sybase will utilize the Intraware SubscribeNet service to provide anytime, anywhere access and delivery of product releases on a global basis. www.intraware.com

S&P Upgrade

12/06/02 – Standard & Poor's Application and Systems Software equity analyst Jonathan Rudy upgraded the stock of Sybase to a "Buy" ranking from a "Hold" based on the integration provider's attractive valuation level. www.standardandpoors.com

ENTERPRISE PORTAL

Digital China

12/03/02 – Sybase announced a strategic alliance partnership with Digital China, China's largest systems integrator, in providing e-business solutions in China. Through the partnership Digital China is expected to become one of the first and biggest portal systems integrators in China. www.digitalchina.com.hk

*i*ANYWHERE

Awards

12/17/02 – *i*Anywhere Solutions announced that the company recently received industry accolades from *Field Force Automation*, *Forbes*, and *InfoWorld*. These publications recognized *i*Anywhere Solutions' leadership in the mobile and remote enterprise computing industry, extending business-critical information to workers anywhere, anytime.

Market Share

12/17/02 – *i*Anywhere Solutions, a subsidiary of Sybase, announced that it leads the mobile database market for the sixth year in a row. Sybase was able to increase its share of the market from 68% in 2000 to 73% in 2001. As the global economy slowly pulls out of recession, the mobile DBMS market is expected to pick up fairly rapidly. Gartner is predicting a compound annual growth rate of over 16% through 2006. This will cause the market to more than double, from slightly under $70 million in 2001 to around $150 million in 2006.

ADAPTIVE SERVER ENTERPRISE

HP

12/16/02 – Sybase and HP announced an expansion to their strategic alliance that will provide customers with greater choice for implementing a high-performance, cost-effective data management solution. The companies plan to provide customers with a total solution comprising Adaptive Server Enterprise running on HP-UX and Red Hat Linux operating systems powered by HP's Itanium 2–based servers. The companies also announced plans for Sybase ASE to be available on HP's forthcoming EV7 processor-based AlphaServer systems. www.hp.com

PeopleSoft CRM 8.8

12/09/02 – Sybase announced their support of PeopleSoft CRM 8.8. PeopleSoft CRM 8.8 includes more than 40 new features and product enhancements to address the market demand for highly scalable systems. PeopleSoft CRM 8.8 expands upon PeopleSoft's functionally rich CRM applications, making CRM easier to use and deploy globally. New features include more than 30 enhancements to its analytics suite, a multichannel module for e-mail and live chat, and a redesigned user interface. www.peoplesoft.com

CYA HOTBackup

12/02/02 – CYA Technologies announced that CYA HOTBackup is now available for use with Sybase's Adaptive Server Enterprise. www.cya.com

INDUSTRY WAREHOUSE STUDIO

Los Angeles Times

12/03/02 – Sybase announced that the *Los Angeles Times* had selected Industry Warehouse Studio for Media to monitor advertising market penetration, product effectiveness, and financial trends in its growing preprint advertising business. www.latimes.com

EVENTS

5th Annual Sybase Tools Seminar

Bloomington, MN
April 7, 2003
Offers even more presenters and room for more attendees! www.powerobjects.com/seminar

TechWave 2003

Orlando, FL
August 3–8, 2003
www.sybase.com/techwave2003 ▼

bruce.armstrong@teamsybase.com

E.Crane Computing

www.ecrane.com

Sybase, Inc

www.sybase.com